

SEGMENTATION ADAPTATIVE POUR LE CODAGE D'IMAGES

THESE No 691 (1987)

PRESENTÉE AU DEPARTEMENT D'ELECTRICITE

ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ES SCIENCES TECHNIQUES

PAR

RICCARDO LEONARDI

ingénieur électricien diplômé EPFL
de nationalité italienne

acceptée sur proposition du jury :

Prof. M. Kunt, rapporteur
Prof. G. Coray, corapporteur
Prof. M. Eden, corapporteur
Prof. H. Maitre, corapporteur
Prof. F. Rocca, corapporteur

Lausanne, EPFL
1987

À mes parents

Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles,
And by opposing, end them.

Hamlet (III,1), Shakespeare.

Remerciements

Je tiens à remercier tout d'abord mon directeur de thèse, le Professeur Murat Kunt. Grâce à lui, j'ai appris à mener à bien un travail de recherche. Pendant trois ans et demi, nous avons étroitement collaboré. Il a su me démontrer l'importance d'organiser son propre travail, le poids qu'il faut mettre dans la présentation de ce dernier pour qu'il puisse être apprécié par d'autres. Par ses nombreux contacts internationaux, j'ai pu connaître la communauté scientifique du traitement du signal, qui je l'espère me sera fort utile tout au long de ma carrière professionnelle. Enfin, je remercie un homme qui restera pour moi un ami.

Je remercie également mes collègues de bureau Michel Bénard et Patrick Volet dont les conseils m'ont toujours été fort utiles. Leur manière très diverse de voir les problèmes n'a pu qu'amener de l'eau à mon moulin.

Mes remerciements vont ensuite à tous ceux qui ont relu le manuscrit de ma thèse, et dont les nombreux commentaires m'ont été fort précieux. En l'occurrence, il s'agit du Professeur Murray Eden et des Docteurs Martin Vetterli et Roberto Cusani. Je leur sais gré du plaisir que j'ai eu à collaborer avec eux et j'espère que nous pourrons continuer à nous aider réciproquement dans nos activités de recherche.

Je remercie aussi les membres de mon jury d'avoir accepté de participer à l'évaluation de ce travail et pour les commentaires utiles qu'ils ont pu m'apporter.

Merci également au Professeur Frédéric de Coulon, Directeur du Laboratoire de Traitement des Signaux pour son regard critique face aux problèmes étudiés (quantification vectorielle).

Je tiens à remercier aussi le Professeur Henri Nussbaumer, qui m'a permis d'accéder aux moyens informatiques de son laboratoire pour la rédaction de ce travail.

Un merci tout particulier à Michel Kocher, qui ayant posé les jalons de la philosophie de ce travail m'a permis d'arriver aux résultats finaux. Souvent, ses conseils et son expérience se sont avérés d'une très grande utilité.

J'adresse enfin mes chaleureux remerciements à tous les membres actuels et passés du Laboratoire de Traitement des Signaux pour leur aide, leurs conseils et leur soutien; en particulier, Michael Unser, Anastasios Ikononopoulos, Pierre-Alain Moret, Jean-Charles Pandazis, Pierre-François Regamey, Erwin Beck, Roger Vuistiner, Jean-Marc Vesin, Mario Blais, Eric Gruaz, Nazih Ziade, Caspar Horne, XiChuen Du, Hans du Buf, Mohammad Kardan, Daniel Cano, Thien Haminh, Gou Youedong, Jie Wang et notre chère secrétaire Heidi Bernau.

Un dernier merci à tous mes amis et membres de ma famille qui ont dû me supporter pendant ces derniers mois.

Résumé

Ce travail de doctorat traite le problème de la segmentation des images en un ensemble de régions correspondant autant que possible aux objets de la scène.

L'objectif de cette segmentation est de passer d'une représentation numérique à une représentation symbolique des images, c'est-à-dire les régions et les paramètres qui les caractérisent. On passe ainsi d'un modèle local constitué par l'ensemble des échantillons numériques de l'image à un modèle plus élaboré où les frontières des régions constituent les contours des objets de l'image et le signal luminance à l'intérieur de chaque région leur texture.

Une telle représentation peut trouver des applications en interprétation de scène et en codage d'images, ceci à cause de la validité du modèle contour-texture utilisé en relation avec les propriétés du système visuel humain. Nous avons étudié le potentiel de la méthode de segmentation dans le contexte de la réduction de redondance appliquée aux images.

La méthode de segmentation proposée procède de manière adaptative. Etant donné une approximation pour modéliser la texture à l'intérieur de chaque région, on cherche à adapter d'une façon optimale les paramètres de cette approximation à la forme des régions. Pour y parvenir, l'image est décomposée dans un premier temps en un ensemble de carrés de différentes tailles de manière à obtenir une correspondance aussi grande que possible entre le signal original et son approximation à l'intérieur de chaque carré. Dans un deuxième temps, ces carrés sont itérativement rassemblés jusqu'à atteindre un certain nombre de régions de forme quelconque. On regroupe successivement les deux régions les plus ressemblantes selon la correspondance qui existe entre le signal original et son approximation sur la région obtenue après rassemblement.

On a étudié les résultats de cette méthode en représentant la luminance de chaque région par des fonctions polynômiales de degré 0 à 3. Une fois la segmentation finale obtenue, les informations de frontières de régions et d'approximation sont codées séparément.

Les performances en compression d'information sont remarquables: des images de qualité tout à fait acceptable peuvent être obtenues avec des taux de compression dépassant 30 à 1. Le contenu sémantique essentiel d'une scène peut être préservé pour des facteurs de compression allant de 60 à 1 à 130 à 1 selon le genre d'image traitée.

Abstract

ADAPTIVE SEGMENTATION FOR IMAGE CODING.

This Ph.D. dissertation analyses the problem of segmenting an image into a set of regions corresponding as much as possible to the real objects of a scene.

The goal of this segmentation is to go from a numerical representation of an image to a symbolic one, i.e. the regions and their characteristic features. Starting from the set of picture elements, one reaches a more compact model where region frontiers define the contours of the objects and the signal within each region their texture.

Such a representation can find useful applications for scene understanding and image coding. Recent works have shown the potential of a contour-texture model for image coding due to the properties of our human visual system. We have studied here the way of coding a segmented representation of an image when using high order polynomials for approximating the different regions.

The proposed segmentation algorithm is adaptive. Given a certain approximation to model the texture within each region, one tries to modify adaptively the region shape and the approximation parameters. To do so, the image is first split into a set of squares of different sizes in order to obtain an optimal correspondance between the original signal and its approximation within each square. Then starting from this initial partition, adjacent regions are iteratively merged till one reaches a segmentation with a certain number of regions of any shape. At each step of the merging process, the two most similar regions are associated on the basis of the adequacy of the approximation on the new region.

Polynomials of degree 0 to 3 have been used in the approximation process. Once the final segmentation is obtained, frontier information and texture information are coded separately.

Performances for redundancy reduction are impressive: acceptable quality images can be obtained with compression ratios of the order of 30 to 1. It is shown how most of the semantics can be preserved with coded pictures at compression ratios ranging from 60 to 1 to 130 to 1. The algorithm has been applied to three different 256x256 natural images quantized with eight bit dynamics.

Table des matières

<u>Remerciements</u>	iii
<u>Résumé</u>	iv
<u>Abstract</u>	v
<u>Tables des matières</u>	vi-x
1 Introduction	1-7
<i>1.1 Segmentation et compression d'images</i>	1-5
1.1.1 Codage d'images à haute compression (méthodes de deuxième génération)	2-3
1.1.2 Historique des méthodes de segmentation	3-5
- Formation de frontières	4
- Formation de régions	4-5
<i>1.2 Contribution de notre travail</i>	5-6
<i>1.3 Organisation de la thèse</i>	6-7
2 Segmentation adaptative	8-45
<i>2.1 Modèle contour-texture</i>	8-21
2.1.1 Buts visés	11-15
- Commentaires	14-15
2.1.2 Approche adaptative	15-16
2.1.3 Adaptation des paramètres du modèle	16
2.1.4 Adaptation de la forme des régions: Division et Rassemblement	16-20
2.1.5 Complexité du problème	20-21
<i>2.2 Extraction d'images de contrôle</i>	22-33
2.2.1 Hypothèses sur les images traitées	22
2.2.2 Extraction de contours	22-32
2.2.3 Images des structures positives et négatives	32-33
<i>2.3 Approximation des données</i>	34-43
2.3.1 Type d'approximation	34-40
- Considérations générales	35-37
- Critère d'approximation minimax	37-38
- Effet du paramètre p	38-39
- Effet des coefficients de pondération $a(k,l)$	39
- Approximation au sens des moindres carrés	39-40
2.3.2 Fonctions d'approximation	40-41

2.3.3 Quantification de l'approximation	42-43
2.4 Mesures de qualité	43-45
2.4.1 Mesures globales	43-44
2.4.2 Mesures contextuelles	44-45
- Commentaires	45
2.5 Conclusion	45
3 Opération de division	46-93
3.1 Principes généraux	46-49
3.1.1 Existence d'une solution dans le calcul de l'approximation	48
3.1.2 Formes des régions après division	48-49
3.2 Implémentation de l'algorithme	49-51
3.2.1 Analyse hiérarchique	50
3.2.2 Analyse descendante-ascendante cyclique	51
3.3 Complexité de l'algorithme de division	51-59
3.3.1 Aspects liés à l'approximation	52-57
- Rôle de la séparabilité	55-57
3.3.2 Simplifications rendues possibles grâce à la structure des images traitées	57-59
3.4 Représentation de l'information segmentée et postraitement	60-67
3.4.1 Position et taille des carrés	60-65
- Codage par arbre quaternaire	60-61
- Codage conditionnel par taille de carré	61-65
3.4.2 Quantification des coefficients polynômiaux	65-66
3.4.3 Postraitement	66-67
3.5 Analyse expérimentale de l'algorithme de division	67-92
- Convention	68
3.5.1 Division effectuée sans image de contrôle des contours	68-72
3.5.2 Effets des paramètres et compression	73-80
- Rôle du seuil sur l'erreur quadratique moyenne	80-84
- Rôle du degré du polynôme	84-86
- Aspects liés à la compression d'information	86-90
3.5.3 Limitation de la plus petite taille de carré	90-92
3.6 Conclusion	92-93
4 Opération de rassemblement	94-139
4.1 Principe de l'algorithme	94-101
4.1.1 Précision dans la description des frontières de région	95-97
4.1.2 Choix des critères de dissimilarité entre régions	97-100
- Mesure de distance entre coefficients d'approximation u_i sur chaque région	97

- Erreur quadratique (moyenne) par rapport au signal original	97-98
- Erreur quadratique (moyenne) entre signal approximé avant regroupement et signal approximé après regroupement	98-100
- Erreur quadratique courante pondérée par le nombre de points de chaque région	100
- Remarque	100
4.1.3 Critères d'arrêt du processus	101
Commentaire	101
 <i>4.2 Evolution de l'approximation après rassemblement</i>	 101-106
4.2.1 Décomposition de l'approximation au sens des moindres carrés	101-102
- Commentaires	103-104
4.2.2 Application à l'algorithme de rassemblement	104-105
4.2.3 Evaluation des critères de dissimilarité	105-106
 <i>4.3 Structure des données et implantation du GCR</i>	 107-112
4.3.1 Réalisation des noeuds du GCR	107-108
4.3.2 Réalisation des branches du GCR	108-109
4.3.3 Mise à jour et ordonnancement des mesures de dissimilarité	109-110
4.3.4 Structure BRAIN	110-112
 <i>4.4 Complexité de calcul et solutions sous-optimales</i>	 113-117
4.4.1 Complexité de l'algorithme	113-115
4.4.2 Simplification du calcul d'approximation	115-116
4.4.3 Solutions sous-optimales par modification du processus de rassemblement	116-117
 <i>4.5 Stratégie globale de rassemblement</i>	 117-138
4.5.1 Analyse des défauts	117-130
4.5.1.1 Analyse expérimentale des effets des critères de dissimilarité	118-127
4.5.1.2 Distorsions causées par l'augmentation du degré du polynôme	128-130
4.5.2 Regroupement des régions sans contours	130-132
4.5.3 Insertion dans le GCR des régions adjacentes aux contours	132-135
4.5.4 Rassemblement final	135-138
 <i>4.6 Conclusion</i>	 139
 5 Représentation de l'information segmentée et postraitement	 140-167
 <i>5.1 Codage de l'information contour</i>	 140-154
5.1.1 Codage de frontière par image interfrontière	143-145
5.1.2 Elimination des liens de régions	145-149
5.1.3 Algorithme de codage du graphe de contours	149-154
 <i>5.2 Codage de l'information texture</i>	 155-164
5.2.1 Coût maximal des coefficients d'approximation	155-157
5.2.2 Quantification vectorielle des coefficients polynômiaux	157-162

- Formulation du problème	157-158
- Algorithmes de coalescence	158-159
- Mesures de distorsions	159-162
5.2.3 Variation du nombre de coefficients par région	162-163
5.2.4 Stratégie globale	163-164
<i>5.3 Elimination des faux contours</i>	165-167
<i>5.4 Conclusion</i>	167
6 Résultats en codage d'images	168-231
<i>6.1 Paramètres d'expérimentation et calcul de compression</i>	168-175
6.1.1 Degré du polynôme	168-169
6.1.2 Nombre de régions	169-170
6.1.3 Autres paramètres intervenant dans la qualité des résultats	170-173
6.1.4 Facteur de compression	173-175
<i>6.2 Résultats expérimentaux et discussion</i>	175-214
6.2.1 Segmentation adaptative par polynômes de degré 0	175-185
6.2.2 Segmentation adaptative par polynômes de degré 1	185-193
6.2.3 Segmentation adaptative par polynômes de degré 2	194-203
6.2.4 Segmentation adaptative par polynômes de degré 3	204-214
<i>6.3 Effet du prétraitement de l'image originale</i>	215-218
<i>6.4 Améliorations</i>	219-231
6.4.1 Segmentation adaptative avec modification du degré de polynôme	219-226
6.4.2 Estimation de la composante aléatoire	227-231
7 Conclusion générale	232-234
<i>7.1 Principe de la segmentation adaptative</i>	232
<i>7.2 Résultats obtenus dans le contexte du codage d'images</i>	232-233
<i>7.3 Problèmes ouverts et améliorations</i>	233-234
<u>Annexe A:</u> Minimisation de la fonction erreur d'approximation $A(e) = e^T A e$	235
<u>Annexe B:</u> Modification de l'approximation au sens des moindres carrés par orthogonalisation de l'espace des fonctions de base	236
<u>Annexe C:</u> Existence d'une solution polynômiale optimale au sens des moindres carrés sur des domaines carrés de dimension paire	237-238
<u>Annexe D:</u> Nombre d'opérations nécessaires au calcul des produits matriciels de l'équation (3.12)	239-240

<u>Annexe E:</u> Estimation de l'erreur quadratique lors d'une formulation séparable du problème d'approximation - Application aux polynômes séparables	241-242
<u>Annexe F:</u> Modification de la liste des voisins après rassemblement	243-244
<u>Annexe G:</u> Expression des paramètres d'approximation par changement d'origine	245-246
<u>Annexe H:</u> Invariance de la distance de Mahanalobis par changement d'origine pour des vecteurs de coefficients polynômiaux	247
<u>Références</u>	248-252
<u>Curriculum Vitae</u>	253

1. Introduction

1.1 Segmentation et compression d'images :

Avec les sons, les images constituent pour l'être humain les moyens les plus importants de percevoir la réalité du monde environnant. La perception de la lumière, puis le sens de la vision se développent dès sa naissance, pour aboutir à un système visuel complexe. Ce système est l'étape d'une évolution de plusieurs millions d'années. Ses performances sont impressionnantes.

Notre rétine reçoit de multiples informations chaque seconde. Ces informations sont combinées, intégrées, comparées tout au long d'une chaîne de traitement qui permet de voir puis d'interpréter la scène qui s'offre à nos yeux. La distinction de formes simples tels lignes droites, angles droits, cercles... s'effectue au niveau des cellules simples, complexes et hypercomplexes du cortex visuel. Cependant, l'architecture et le fonctionnement de notre système nerveux nous sont encore trop inconnus pour qu'on puisse comprendre comment s'effectue la phase d'interprétation en elle-même.

Aussi, il est utopique de vouloir réaliser un système de vision par ordinateur qui puisse avoir des performances comparables à celles de notre système visuel. Si l'homme ne peut encore connaître le fonctionnement neurophysiologique de son cerveau, la nature lui a donné une capacité créatrice à partir de ce qu'il perçoit. Il sait analyser, comprendre, modéliser, inventer... D'après les grands philosophes grecs tels Platon, il existe au niveau absolu un concept, une idée. Pour comprendre le fonctionnement du monde, l'homme bâtit des modèles qui vont se rapprocher tant que possible de cette idée. Le modèle peut se situer au niveau de la représentation de la réalité, en l'occurrence l'image. Il peut se faire en termes d'objets connus. Un modèle de la connaissance est nécessaire pour caractériser les objets détectés dans une image. Ces opérations se situent au niveau de ce qui est appelé le traitement d'images de haut niveau. Dans un premier temps, il est cependant nécessaire de décomposer l'image en un ensemble de régions qui correspondent plus ou moins à des objets. Cette opération que l'on appellera la segmentation fait partie du traitement d'images de bas niveau. Elle constitue l'une des étapes fondamentales de l'analyse de scène.

Une segmentation idéale associe à chaque objet de la scène une région. En réalité, celle-ci ne peut être obtenue qu'une fois que l'on a compris le contenu de la scène. Dans un traitement de bas niveau, il s'agit plus d'associer des parties d'objets correspondant à des surfaces homogènes pour constituer une première segmentation. C'est l'un des objectifs de ce travail. Plus précisément, on va chercher à créer une représentation d'une image en un ensemble de régions dont on peut modéliser de façon plus ou moins approchée la luminance par un certain nombre de paramètres. La partition de l'image se fera par une adaptation conjuguée de la valeur de ces paramètres et de la forme de chaque région.

A partir d'une telle représentation, on peut essayer d'interpréter le contenu de l'image. Il apparaît cependant ambitieux de vouloir combiner dans un seul travail le traitement d'image de haut et de bas niveau. Cependant, les symboles que constituent les objets ou les parties d'objet d'une image obtenus par segmentation constituent une représentation très compacte de l'information. Pourquoi ne pas essayer de trouver des applications de la segmentation à la compression

d'information dans les images? A la suite de travaux récents en codage d'images [1], il est apparu qu'on pouvait réduire l'information à transmettre de plusieurs ordres de grandeurs en adoptant une représentation contour-texture d'une image [2], tout en conservant l'essentiel du contenu informationnel de l'image. En définissant une méthode de segmentation qui s'inscrit parfaitement dans le cadre de ce modèle, ses performances en codage d'images se doivent d'être analysées vu l'impact toujours considérable de ce sujet.

1.1.1 Codage d'images à haute compression (méthodes de deuxième génération):

Le but du codage d'images est de minimiser autant que possible la quantité d'information (nombre de bits) nécessaire à une représentation fidèle d'une image originale. Quel que soit le système d'acquisition d'une image, la représentation numérique qui en résulte (appelée forme canonique de l'image) correspond à une suite d'échantillons de luminance régulièrement espacés et quantifiés en un certain nombre de valeurs appelées niveaux de gris. Le pas d'échantillonnage est en général choisi suffisamment petit pour éviter d'interpoler les échantillons lors la reconstitution du signal analogique, c'est-à-dire l'affichage. Il utilise ainsi à bon escient la capacité d'intégration de l'oeil humain. Une image numérique constitue ainsi un tableau bidimensionnel de nombres.

Le contenu de ce tableau n'est en général pas aléatoire. En effet, des échantillons voisins ont des valeurs comparables, ce qui correspond à une importante corrélation spatiale. En exploitant cette corrélation, on peut représenter d'une façon compacte ces valeurs. La plupart des techniques de codage d'images dites de première génération [3] ont ainsi essayé de tirer profit de cette corrélation pour maximiser la compression d'information.

En se limitant à cette approche, on ne s'est pas placé pour autant au niveau de l'entropie de l'image qui est souvent inconnue et dépend fortement du modèle utilisé. En effet, les techniques classiques de codage admettent comme implicite le fait que l'on ait à faire à des signaux stationnaires et ergodiques, chose qui est rarement satisfaite dans le contexte d'images réelles. Si les méthodes de première génération ont mis l'accent sur la façon de coder les messages extraits d'une image, aucun effort n'avait été porté alors quant au choix de ces messages. Les méthodes de deuxième génération ont mis l'accent sur ce point.

Les images traitées ou codées par ordinateur sont en général destinées à un observateur humain. Même si ce n'est le cas, ne faut-il pas essayer de réaliser grâce à l'informatique des analyses ou des traitements comparables à ceux qu'effectue notre système visuel? Au lieu d'associer les échantillons d'une image numérique arbitrairement pour prédire la valeur de l'échantillon suivant ou pour calculer un ensemble d'attributs (coefficients transformées) sur cette série d'échantillons, ne peut-on pas dissocier contours et textures d'une image, comme l'ont proposé Kocher et Kunt [2]? Les contours correspondent aux endroits de variation brusque de luminance. Selon la rapidité du saut de luminance et son amplitude, il est possible de séparer les contours en contours faibles et contours forts. Quant aux textures, la notion correspondante est liée à la perception visuelle et elle relève de beaucoup d'aspects subjectifs [4]. Il n'existe pas de définition communément admise les caractérisant. On peut admettre qu'une texture est invariante par translation [5]. Dans le contexte du codage d'images, les textures représentent tout ce qui n'est pas contour. Réaliser une compression élevée peut

être obtenu à partir d'une décomposition d'une image en termes de régions à texture uniforme entourées par des contours. Si les contours sont fermés, on aboutit à un modèle segmenté de l'image tel celui de Kocher et Kunt [2], modèle dans lequel s'inscrit notre travail et qui sera élaboré au chapitre 2.

Si les frontières des régions ne sont pas fermées, on peut décomposer l'image en une composante basse-fréquence caractérisant l'information de texture et une composante haute-fréquence. Cette dernière peut être codée telle quelle en utilisant la réponse impulsionnelle du filtre haute fréquence; il s'agit de la méthode des "Intensités Artificielles" [6]. Une autre possibilité consiste à séparer la partie haute fréquence en une série de filtres directionnels [7,8]. Ceux-ci vont permettre de caractériser les contours de l'image suivant leur orientation. Les contours dans chaque direction peuvent être sous-échantillonnés vu qu'ils ont été détectés par des filtres directionnels limités en bande. Le profil du signal haute fréquence de part et d'autre de chaque contour est modélisé par des fonctions paramétriques unidimensionnelles à 2 ou 3 variables (polynômes, gaussienne) ou par l'amplitude du profil et la réponse impulsionnelle du filtre somme des filtres directionnels, c'est-à-dire le filtre haute-fréquence (Intensités Artificielles).

Carlsson a proposé récemment une technique où il est possible de reconstruire une image en se basant uniquement sur l'information de contour [9]. Dans un premier temps, les contours importants sont détectés en utilisant un masque Laplacien adouci. L'information de niveau de gris de part et d'autre de ces contours ainsi que leur position sert au codage. Dans toutes les zones ne correspondant pas à des contours, on reconstitue le signal en résolvant une équation de Laplace par une méthode itérative de relaxation. Les conditions aux limites sont imposées par la valeur des niveaux de gris situés de part et d'autre des contours. La résolution d'une équation de Laplace est équivalente à imposer un minimum de variation de luminance dans les zones de texture. Le résidu, c'est-à-dire la différence entre l'image originale et l'image reconstruite est codé par codage pyramidal [10].

Après avoir placé le contexte des techniques de codage de deuxième génération basées sur un modèle contour-texture des images, faisons un historique des méthodes de segmentation.

1.1.2 Historique des méthodes de segmentation:

Un large nombre de techniques de segmentation ont été proposées tout au long du développement des méthodes d'analyse de scène. La plupart des approches se sont limitées à des applications particulières (par exemple, [11,12]) ou se sont fixées des buts bien définis (par exemple, [13]). Il n'existe encore aucun système capable de caractériser les objets de n'importe quelle image naturelle d'une façon parfaite.

On distingue les méthodes de segmentation dites de formation de frontières des méthodes de formation de régions. Les algorithmes permettant d'extraire les frontières impliquent des opérations de différentiation, de recherche de maxima et de regroupement de segments locaux en contours. Les approches visant à former des régions comprennent la croissance de région par analyse locale, l'analyse de propriétés globales.

Formation de frontières:

En appliquant des opérateurs de différentiation spatiale à une image, on peut mesurer l'importance de son gradient. En plaçant un seuil sur ce dernier, on obtient une image formée d'un ensemble de contours séparés. Si cela s'avère nécessaire, il faut éliminer les contours peu importants ou les contours fictifs (c'est-à-dire ceux qui correspondent à des réponses multiples de l'opérateur de différentiation utilisé). Les contours restants sont alors associés pour représenter les frontières de régions.

Une grande partie des débuts de la recherche en analyse de scène était basée sur la recherche d'algorithmes de poursuite de lignes droites [14,15]. En se limitant à des images composées d'objets pouvant être assimilés à des polyèdres, plusieurs auteurs ont proposé d'utiliser la position des sommets de ces polyèdres pour relier les différents segments détectés [16-18]. La contrainte imposée de n'avoir que des formes polygonales rendait l'utilisation seule de ces méthodes inefficace pour la segmentation d'images naturelles. Mackworth [19] puis Horn [20] ont essayé de baser la définition de leurs algorithmes de poursuite de frontières sur une analyse des surfaces de luminance (leur orientation, leur courbure, la lumière qu'elles réfléchissent). Il est possible de poursuivre un contour en recherchant les points de valeur comparable de gradient. Certains auteurs ont proposés d'extraire globalement certaines informations relatives aux contours pour pouvoir faciliter leur association [21,22]. Ainsi, des contours possédant des propriétés directionnelles similaires (estimées globalement) peuvent être utilisées au moment d'une analyse locale pour effectuer la formation de frontières.

Zucker, Hummel et Rosenfeld ont suggéré [23] d'utiliser des procédures de relaxation pour la formation de frontières. L'avantage de cette méthode consiste à considérer toutes les orientations possibles en différents points contours voisins d'un certain point contour pour définir l'orientation la plus probable du contour en ce point.

Quelle que soit la démarche choisie, aucune méthode de formation de frontières ne peut donner naissance à des frontières ayant toutes une signification sémantique au niveau de l'interprétation. Il en va de même au niveau des régions obtenues par les techniques de formation de régions que l'on va décrire ci-dessous.

Formation de régions:

Il est possible de décomposer n'importe quelle image en un ensemble de régions en regroupant tous les points images qui possèdent une certaine propriété avec une tolérance n'excédant pas un certain seuil. On utilise en général un balayage successif de tous les points voisins d'une certaine région pour les inclure à cette région. Cette approche générale est connue sous le nom de croissance de région. Le seuil de tolérance ne peut être fixé une fois pour toute pour une certaine image si l'on désire obtenir des régions se rapprochant des objets de la scène. Toutes sortes de solutions ont été proposées pour fixer le seuil de tolérance de façon dynamique mais il est difficile de le faire si l'on n'estime pas des paramètres plus globaux et si l'on n'a pas de connaissance a priori de la scène que l'on désire segmenter. Freuder [24] a introduit une méthode relativement intéressante pour

regrouper des régions en effectuant une mesure de similarité entre régions voisines et en décidant de regrouper les deux qui sont les plus ressemblantes. Plus tard, Horowitz et Pavlidis [25] ont suggéré de limiter l'analyse de toutes les configurations possibles de régions à comparer deux à deux en introduisant le concept de division et rassemblement. L'image est subdivisée dans un premier temps en zones homogènes de forme simple (par exemple, des carrés de différentes tailles). Ces carrés sont alors comparés pour être rassemblés jusqu'à obtenir des régions de forme quelconque.

Une approche plus récente encore consiste à effectuer la formation de régions sur la base d'une analyse sémantique. A cet égard Levine et Nazif [26] ont construit un système expert. A notre avis, les résultats obtenus restent cependant comparables aux méthodes de croissance de région [27] ou de division et rassemblement [25] classiques.

Certains auteurs ont enfin suggéré de former des régions en effectuant des mesures de caractéristiques d'une image. Olhander [28] montre ainsi comment découper itérativement une image en plaçant des seuils aux minima des histogrammes de différents attributs extraits de l'image. Malheureusement, ce genre de méthode ne tient pas bien compte des relations spatiales des différentes régions. A cet effet, Hanson, Riseman et Nagin [29] ont décrits comment les prendre en considération en estimant les différents attributs sur des fenêtres de différentes tailles.

Analysons maintenant en quoi notre travail se distingue des méthodes présentées jusqu'ici.

1.2 Contribution de notre travail :

Notre travail aborde en premier lieu le problème de la segmentation d'une scène naturelle. La méthode de segmentation proposée s'intègre dans le contexte des méthodes de formation de régions. Elle introduit cependant le concept de *segmentation adaptative*. Dans un premier temps, on définit un modèle souple basé sur des approximations polynomiales pour représenter différentes configurations de luminance. Dans un deuxième temps, on effectue un arrangement optimal entre les paramètres du modèle polynomial servant à décrire le signal luminance à l'intérieur de chaque région et la forme de ces régions. C'est à ce niveau qu'apparaît l'adaptation. L'algorithme de formation de régions se base sur la philosophie de division et rassemblement introduite par Horowitz et Pavlidis [25].

En montrant l'importance de l'information de contour dans les images, des contraintes sont imposées à la méthode de segmentation de manière à préserver ce genre d'information. Dans la segmentation finale, les frontières des régions vont épouser autant que possible les contours de l'image. De ce fait, les régions obtenues se rapprochent d'objets ou de parties d'objets constituant la scène. L'essentiel du contenu sémantique de la scène est ainsi préservé. Il est donc tout à fait envisageable de comparer les régions obtenues à des objets connus pour effectuer une interprétation de l'image. Celle-ci ne sera pas abordée ici mais un mode de représentation des frontières de régions et du modèle d'approximation de luminance va permettre d'utiliser les résultats de la segmentation dans le contexte du codage d'images.

A cause du grand nombre de paramètres que l'on peut utiliser pour aboutir à la segmentation finale, la méthode de codage proposée s'avère efficace pour représenter avec plus ou moins de distorsion une scène originale. En limitant le nombre de régions de la segmentation finale, on augmente la compression mais on élimine les objets de petite taille de l'image. Des résultats de bonne qualité sont obtenus pour des images originales 256x256 à 8 bits comprimées d'environ un facteur de compression de 30 à 1. Dans certains cas, pour des images constituées de grandes plages uniformes, l'information peut être réduite d'un facteur 130 à 1 sans que le contenu essentiel de l'image ne soit perdu et tout en garantissant une précision absolue au niveau des contours réels des objets apparaissant dans l'image segmentée.

Ce travail définit donc une méthode de segmentation adaptative qui garantit une bonne correspondance entre les objets réels d'une scène et les régions formées. La sophistication du modèle contour-texture proposé permet d'utiliser les résultats de la segmentation dans le contexte du codage d'images de moyenne à haute compression selon le degré de détail désiré. On pourra comparer les résultats obtenus aux autres méthodes de codage d'images de deuxième génération (voir paragraphe 1.1.1).

1.3 Organisation de la thèse :

Le chapitre 2 présente la philosophie de la méthode de segmentation adaptative. Il introduit le modèle contour-texture et définit les objectifs visés par l'algorithme de segmentation dans le contexte du codage d'images. Il présente comment adapter de façon conjuguée les paramètres du modèle de représentation de chaque région à la forme de ces régions. Le tout aboutit à un algorithme adaptatif de division et rassemblement dont les détails sont mis en évidence aux chapitres 3 et 4. Ce chapitre décrit également comment extraire des images de contrôle liées aux contours servant à piloter le processus de segmentation. Il justifie le choix du critère des moindres carrés pour estimer les paramètres d'approximation polynomiale du signal à l'intérieur de chaque région. Enfin, il propose des mesures servant à faire évoluer la segmentation jusqu'à obtenir le résultat désiré.

Le chapitre 3 aborde le détail de la première phase de l'algorithme de segmentation: la division adaptative de l'image en des carrés de différentes tailles sur la base d'une correspondance optimale entre le signal d'approximation à l'intérieur de chaque carré et l'original. L'accent est mis sur la non dégradation de l'information contour. Ce chapitre présente également des détails d'implantation quant à la structure des données et à la complexité algorithmique de l'opération de division. Il montre comment représenter sous forme d'un arbre quaternaire l'information à ce niveau de la segmentation. Des résultats préliminaires de codage donnent une idée de la réduction de redondance et de la qualité des images qu'on est en mesure de reconstituer après division.

Le chapitre 4 décrit la deuxième phase de l'algorithme de segmentation: le rassemblement adaptatif des régions obtenues après division. En construisant un graphe permettant de comparer toutes les associations possibles de deux régions adjacentes, on choisit de regrouper d'une façon itérative les deux régions qui introduisent le moins de distorsion au niveau de l'approximation après

regroupement. Pour pouvoir estimer la distorsion introduite, différentes mesures d'erreur sont proposées. Du fait d'une propriété de décomposition de l'approximation au sens des moindres carrés, on montre comment calculer ces mesures rapidement de manière à rendre possible l'implantation de l'algorithme de rassemblement. Comme au chapitre 3, des problèmes de structure de données et de complexité algorithmique sont abordés dans ce chapitre. Enfin, ce chapitre définit à partir d'une procédure expérimentale comment combiner les mesures de distorsions pour s'approcher au mieux d'une segmentation idéale constituée uniquement d'objets de la scène.

Dans le chapitre 5, on décrit la manière de coder chaque type d'information à partir d'une image segmentée, c'est-à-dire information contour et information texture. Les frontières des régions vont être décrites sans erreur au moyen d'un code de chaîne après avoir légèrement modifié la segmentation de manière à imposer des contraintes à la succession des symboles pouvant exister dans le code. L'information texture est caractérisée par des surfaces de reconstruction polynômiales. On montre comment coder les coefficients de ces polynômes. La même surface de reconstruction peut se rapporter à des parties non connexe de l'image grâce à l'utilisation de techniques de quantification vectorielle. Enfin, ce chapitre illustre comment améliorer la qualité de l'image reconstruite en éliminant les faux contours.

Le chapitre 6 donne les résultats de la méthode dans le contexte du codage d'images. Des images codées avec des facteurs de compression allant de 30 à 1 jusqu'à 130 à 1 y sont présentées. On analyse l'effet du nombre de paramètres d'approximation (c'est-à-dire le nombre de coefficients polynômiaux) sur la qualité de la segmentation finale. On montre comment on peut encore améliorer cette qualité en combinant des approximations par des constantes avec des approximations de degré plus élevé. Ce chapitre discute également de l'effet du prétraitement de l'image originale par des opérateurs non linéaires d'élimination de granularité tel que le filtre de Nagao.

Le chapitre 7 sert de conclusion à ce travail. Il montre les problèmes qui restent ouverts en analyse et interprétation de scène et suggère des améliorations qui pourraient augmenter la qualité des résultats obtenus.

Le lecteur désireux d'avoir un aperçu des résultats que l'on peut obtenir peut se limiter aux chapitres 2, 5 et 6. Le chapitre 2 donne les bases de la méthode de segmentation. Le chapitre 5 indique comment coder l'information segmentée alors que le chapitre 6 décrit et discute les résultats de codage. Les chapitres 3 et 4 mettent en évidence l'implantation de l'algorithme de segmentation proposé. Ils décrivent le détail de propriétés mathématiques nécessaires à la réalisation de la méthode ainsi que comment combiner tous les paramètres pour aboutir au résultat désiré.

2. Segmentation adaptative

Ce chapitre présente une méthode assez générale et adaptative de segmentation d'images de manière à faire ressortir les structures de la scène. Il s'agit essentiellement de définir les critères permettant de caractériser une image et donc de la modéliser. L'accent sera mis sur la nécessité d'une analyse contextuelle tenant compte des propriétés statistiques et structurales de l'image. Trois idées essentielles sont à retenir :

- l'image est représentée par un *ensemble de régions*. A chacune de ces régions est associée une certaine approximation permettant de décrire la variation de son niveau de gris.

- la forme de ces régions est définie par l'*adaptation optimale* des paramètres d'approximation à la structure des données.

- la nécessité d'une *approche évolutive* pour déterminer la taille, le nombre et la forme des régions, vu que l'analyse de l'image se fait à toute une série de niveaux pour lesquels les règles sémantiques sont différentes.

Ce chapitre se divise en cinq sections : La première établit un modèle des images et en justifie le choix d'une représentation segmentée. La deuxième traite de l'extraction d'images de contrôle à partir de la scène originale, images qui permettront de piloter le processus de segmentation. La section suivante analyse le problème de la représentation des données pour chaque région et insiste sur la nécessité d'effectuer la partition de l'image en tenant compte du choix qui a été fait pour approximer les données. La quatrième section définit les mesures de qualité qui seront utilisées pour déterminer l'adéquation entre la représentation choisie et le signal de départ; l'accent sera porté sur la nécessité d'une analyse à la fois globale et contextuelle. Enfin, le chapitre se termine par un rappel des points essentiels présentés.

2.1 Modèle Contour-Texture :

Une fonction réelle positive de deux variables x et y , notée $g(x,y)$, définit une représentation mathématique d'une image à niveaux de gris. Sa valeur au point (x,y) , dite niveau de gris, est proportionnelle à l'intensité lumineuse en ce point. En fait, n'importe quel moyen permettant de caractériser l'évolution de cette fonction réelle sur l'étendue finie de l'image constitue un modèle de celle-ci. Nous appellerons "image naturelle" une scène habituelle du monde qui nous entoure, telle un paysage de montagne, un portrait, une ville... Néanmoins, il est souvent difficile, voire impossible de trouver une expression analytique permettant de décrire l'évolution de n'importe quelle image naturelle. Une représentation plus courante et relativement simple consiste à mettre l'image sous sa forme numérique. Il suffit pour cela d'échantillonner la fonction $g(x,y)$ en un certain nombre de points et de quantifier la valeur échantillonnée selon un nombre fini d'amplitudes entre le niveau 0 (noir) et le niveau maximum (blanc). Pour éviter toute perte d'information, les images doivent être échantillonnées en respectant le théorème de Shannon, c'est-à-dire que la fréquence d'échantillonnage soit au moins deux fois supérieure à la fréquence maximale du signal image dans chaque direction. Bien entendu, on peut être moins restrictif en supposant que les effets de recouvrement du spectre

du signal image ne tombent pas dans la bande passante de l'oeil humain, pour autant que l'image soit destinée à un tel observateur. Par la suite, on ne considèrera que des images mises sous forme numérique $g(k,l)$. La grille d'échantillonnage sera supposée rectangulaire et constituée de M lignes \times N colonnes. Le processus de numérisation est présenté à la figure 2.1 (les signaux sont présentés à une dimension pour simplicité).

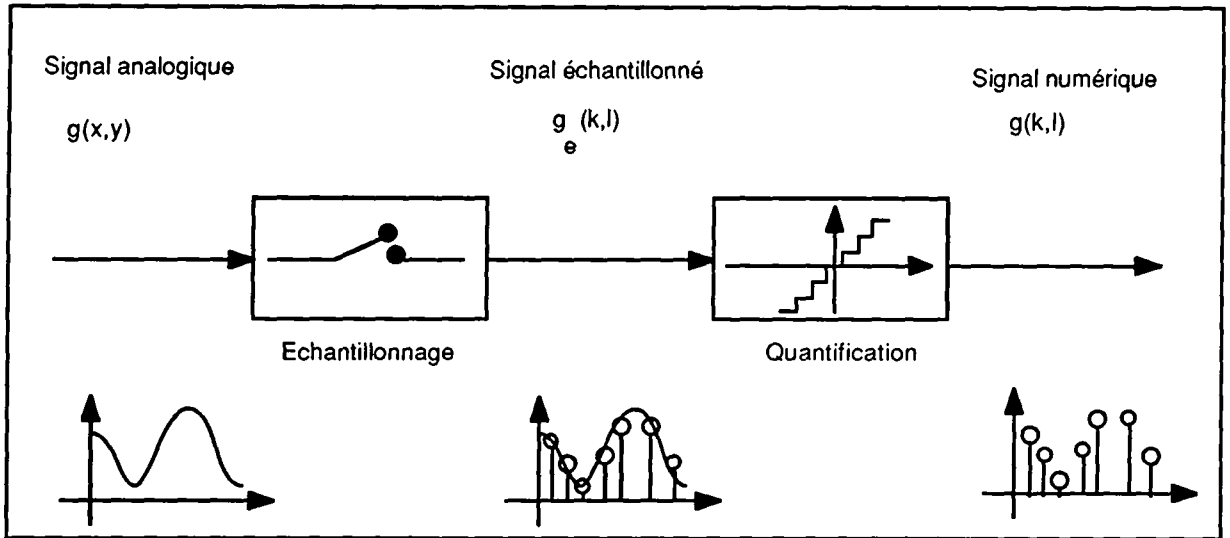


Figure 2.1 : Processus de numérisation d'une image
(les signaux sont présentés à une dimension pour des raisons de simplicité)

Les figures 2.2, 2.3 et 2.4 présentent les trois images naturelles 256x256 à 8 bits qui seront utilisées dans le cadre de ce travail. L'image 2.2 (bâtiment) a été choisie parce qu'elle présentait des caractéristiques géométriques intéressantes. L'image 2.3 (caméraman) possède une partie texturée (herbe) et une partie dont l'évolution du niveau de gris est relativement complexe (caméra). De plus, cette image est très contrastée et présente des détails aussi bien dans les parties claires que dans les parties sombres. L'image 2.4 (couple) est intéressante car elle représente des visages humains. Vu l'habitude que nous avons à côtoyer des êtres humains chaque jour, il est raisonnable de penser que nous sommes particulièrement sensibles aux distorsions apportées à ce genre d'images. Ces images ont été choisies également parce qu'elles permettront de mieux apprécier l'évolution des résultats de codage par rapport au travail de Kocher [27].



Figure 2.2 : Image du bâtiment.

Figure 2.3 : Image du caméraman.

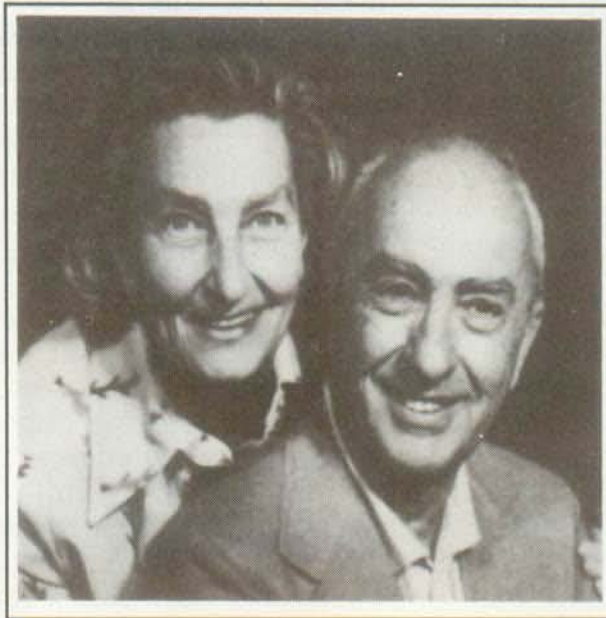


Figure 2.4: Image du couple

La représentation numérique $g(k,l)$ peut déjà être considérée comme un modèle dont les paramètres sont les points images ou pixels. Ce modèle a l'avantage d'être très général mais possède le grave inconvénient de faire intervenir un nombre énorme de paramètres (les pixels), chacun très local.

La perception humaine, en revanche, utilise d'autres paramètres. Par exemple, un dessinateur commence son dessin en esquissant les pourtours des objets et remplit ensuite les régions ainsi délimitées. Les travaux récents de neurophysiologie [30] montrent clairement l'importance du rôle joué par les contours dans la perception. Il est donc légitime d'utiliser un modèle général de contour-texture. Ici, la texture fait référence à tout ce qui n'est pas contour. On peut

y ajouter éventuellement une troisième composante, la structure, pour lier les régions les unes aux autres.

Puisque les contours sont des paramètres de ce modèle, il faut les extraire d'une façon ou d'une autre à partir des données $g(k,l)$. Il existe deux approches générales à l'extraction de contour : celle qui vise à les obtenir directement et celle qui cherche à déterminer les régions de texture uniforme dont les frontières définissent les contours.

Kocher, en adoptant la deuxième méthode [27] a montré son potentiel énorme pour la compression d'information. Toutefois, la généralisation directe de cette approche pour en augmenter davantage les performances, conduit à une complexité très déraisonnable. C'est pourquoi, notre approche se démarque de celle de Kocher et cherche à atteindre les buts suivants.

2.1.1 Buts visés :

La vision par ordinateur simule le fonctionnement de notre système visuel. Il s'agit de trouver les objets d'une image et de caractériser ceux-ci selon une certaine approximation. Cela ne correspond-il pas au processus quotidien de chacun d'entre nous lorsqu'il regarde une scène naturelle? Comme indiqué par Riseman et Arbib [31], la complexité impliquée par sa conception et sa réalisation a conduit à décomposer le problème en toute une série de démarches dites de "bas niveau" et de "haut niveau" (voir chapitre 1). La partition de l'image en une série de régions, c'est-à-dire la segmentation, constitue une étape préliminaire à l'opération d'interprétation. En effet, celle-ci nécessite le plus souvent la manipulation d'entités, issues d'une description par régions.

Comment peut-on définir une stratégie pour passer d'une représentation sous forme de signal numérique à une représentation symbolique d'une scène? Ce problème très général d'analyse de scène peut être résolu de manière satisfaisante lorsque l'on connaît la scène à analyser donc les objets qu'elle contient. Le problème devient beaucoup plus complexe si l'on n'a aucune connaissance a priori de ce que cette scène représente. Le processus de reconnaissance par une machine ou un être humain nécessite un apprentissage, la caractérisation du monde à interpréter au moyen de règles ou de termes connus. La modélisation implique toute une hiérarchie de niveaux de traitement où les symboles sont manipulés selon des règles d'inférence. Il s'agit de définir ces symboles, c'est-à-dire un alphabet simple permettant de caractériser des groupes de symboles d'un niveau d'abstraction (hiérarchique) inférieur.

Pour obtenir une segmentation, on a cherché à modéliser mathématiquement ou statistiquement les variations spatiales de luminance. On désire extraire des paramètres de l'image pouvant définir des invariants. A partir de là, il est associé à chaque région une série d'attributs. Ceux-ci correspondent souvent, du moins en partie, à des entités ayant une signification sémantique, telles des caractéristiques de surface d'un objet. L'interprétation se fait par interaction entre ces différents attributs selon un modèle de connaissance. Le processus d'interprétation permet le plus souvent de parfaire la partition de l'image.

S'il est possible de faciliter l'interprétation d'une image à partir d'une représentation segmentée de celle-ci, une segmentation permet aussi de réduire

considérablement la redondance d'information présente dans une image. Son utilisation dans le contexte du codage d'images est donc souhaitée. En vue de cette application, il est important de souligner les critères nécessaires à une compression élevée sans perte sensible d'information sémantique. En liaison avec la représentation segmentée de l'image, il faut :

- reproduire le plus fidèlement possible les contours réels des objets, c'est-à-dire décrire les frontières des régions qui leur correspondent avec une grande précision.
- admettre une certaine tolérance dans la représentation du signal image à l'intérieur de chaque région.
- minimiser le nombre de régions de manière à éviter de coder des frontières artificielles et un nombre de paramètres d'approximation trop élevé. A chaque région est associée une certaine approximation.
- éventuellement ajouter une composante aléatoire à l'image codée après segmentation de façon à pouvoir lui redonner un aspect plus naturel.

Voyons donc les raisons de ces exigences. On va montrer la nécessité de *préserv*er l'information contour en liaison avec les propriétés du système visuel humain. Un modèle possible du système visuel humain est présenté à la figure 2.5 [1].

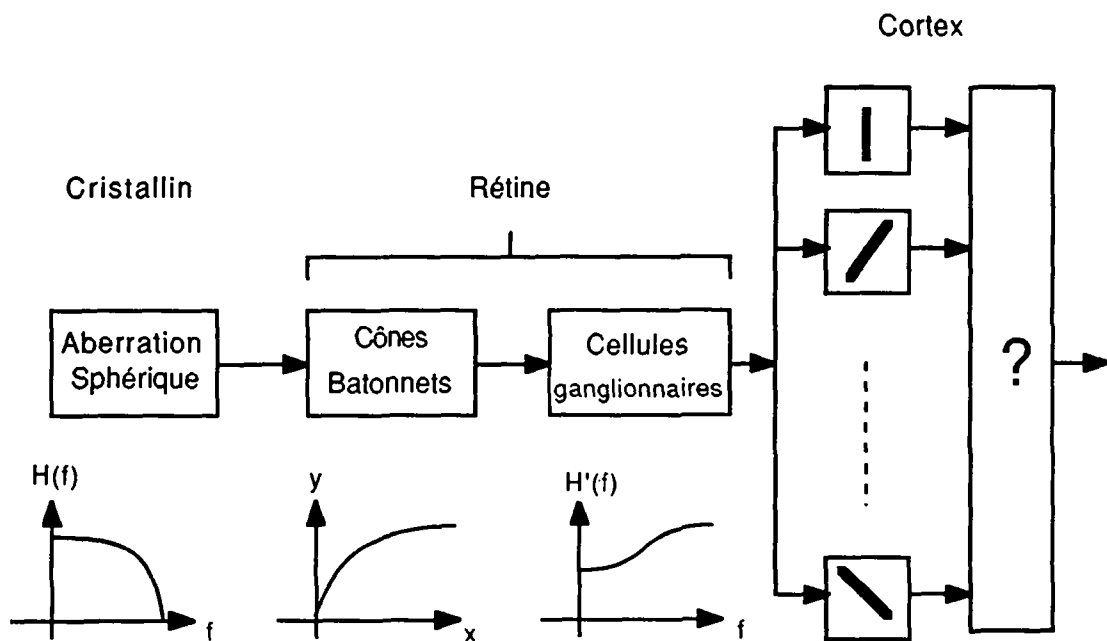


Figure 2.5: Principe de fonctionnement du système visuel humain

Dans un premier temps, on peut considérer en première approximation que l'image est filtrée passe-bas de manière isotropique. Le phénomène physique à la base de ce filtrage correspond à l'aberration sphérique du cristallin. La sensibilité des photorécepteurs (cônes et bâtonnets) au signal image se formant sur la rétine est de type logarithmique. Enfin, le processus d'inhibition latérale des cellules

ganglionnaires agit comme un filtre passe-haut mettant en évidence l'information contour. Au niveau du cortex, cette information est séparée selon différentes directions. Ce modèle très simplifié peut être élaboré à partir des travaux des deux neurobiologistes Hubel et Wiesel [30]. Il permet de comprendre l'importance du contenu sémantique associé aux contours, c'est-à-dire les endroits de variation brusque du signal image. Sans aller jusqu'à analyser la structure même du système visuel humain, on peut se rendre compte de ce phénomène en comparant une scène et l'image des contours associée (Voir figure 2.6). L'information sémantique liée à la scène est bien présente au niveau de l'image des contours.

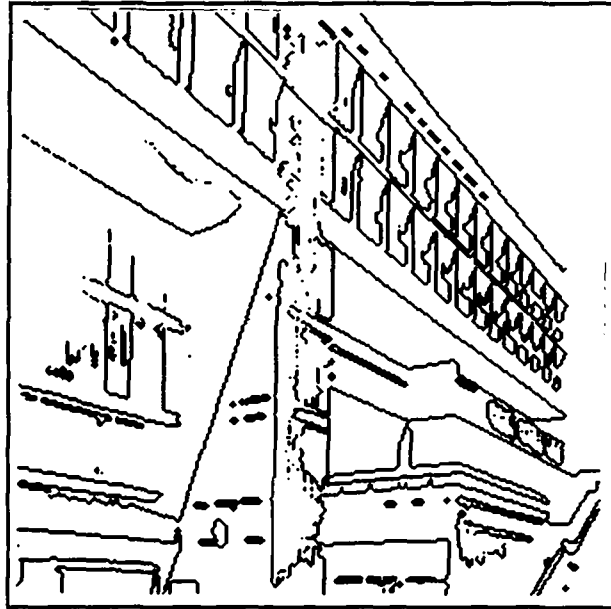


Figure 2.6 : Image des contours du bâtiment

De manière à limiter la dégradation d'une image quant à son contenu sémantique, il faut absolument préserver l'information contour après segmentation, autrement dit :

- éviter de perdre des contours de l'image originale,
- ne pas insérer de contours artificiels, c'est-à-dire inexistant dans l'image originale.

Le non-respect de cette règle résulte le plus souvent en une distorsion parfois irréversible pour l'interprétation. Il est difficile de mesurer objectivement la dégradation subie lors de la modification de l'information contour car son effet est très dépendant de la structure de la scène analysée. On peut néanmoins dire que des contours isolés sans "régularité" correspondent le plus souvent à des composantes aléatoires du signal et sont donc peu porteurs d'information sémantique.

Une zone est dite homogène si tous ses éléments vérifient une même propriété définie à l'avance. La segmentation est l'opération qui permet d'extraire les régions homogènes de l'image. Bien entendu, les frontières de ces régions, si elles correspondent à des contours réels de l'image, doivent être décrites avec la plus grande précision de manière à épouser ceux-ci. Aucune approximation de

frontière ne saurait être tolérée sans avoir analysé la dégradation qui en résulterait au niveau du contenu de l'image.

Quant aux raisons de la deuxième exigence, c'est-à-dire l'approximation des données sur la région, il faut réaliser qu'une compression importante de l'information ne peut être obtenue qu'avec une approximation. Comme celle-ci est interdite au niveau des frontières, elle ne peut intervenir qu'au niveau des données images.

La troisième exigence est liée à l'importance de la compression. En effet, la représentation de chaque région implique la description de ses frontières et de l'approximation du signal luminance qu'elle renferme.

La quatrième exigence se justifie à cause de l'aspect souvent très basse fréquence des régions de grande taille où le modèle d'approximation ne permet pas de reproduire certaines granularités existant dans le signal original.

Commentaires:

1) L'approche de Riseman et Arbib [31] qui consiste à obtenir une segmentation d'une image pour en faciliter son interprétation est controversée. En effet, ne faut-il pas tout de suite essayer d'utiliser la sémantique pour obtenir une bonne segmentation? Le contexte local dans une image influence directement l'interprétation qui en est faite [32]. Tenenbaum et al. [33] reprochent à une modélisation mathématique ou statistique du signal luminance seul, de présenter des limitations considérables. Ceci, car ce genre de caractérisation ne tient absolument pas compte de la structure physique de la scène qui est analysée ni de la manière dont a été formée l'image. Ils opposent ainsi une représentation de l'image obtenue à partir d'un "modèle structurel" (qui tient compte des réalités physiques de la scène) à celle obtenue à partir d'un "modèle statistique". D'après eux, une approche statistique ne se justifie que pour modéliser la composante aléatoire du signal, c'est-à-dire le bruit ou les textures aléatoires. Par contre une approche structurelle s'impose en relation avec l'attitude de l'être humain lorsqu'il regarde une scène. Ce qu'un homme voit au premier coup d'oeil, ce sont les caractéristiques physiques des surfaces des objets qui lui sont présentées indépendamment du point de vue choisi ou de la position de la source lumineuse.

Nous pensons pour notre part que les autres approches ont leur raison d'être soit parce qu'elles s'appliquent à un contexte spécifique où des études simples des caractéristiques du signal luminance correspondent à une réalité physique sous-jacente (ex: détection d'objets plans dans de bonnes conditions d'éclairage), soit parce que la définition d'un modèle de connaissance général d'un système de vision est très difficile à mettre en oeuvre sans utiliser une représentation symbolique justement donnée par la segmentation.

2) Le signal image est décrit pour chaque région de manière relativement simple au moyen d'une approximation. L'aspect modélisation du signal luminance est donc conservé tout en insérant des considérations structurelles par une approche adaptative pour aboutir à la segmentation. Le modèle permettant de décrire le signal luminance est suffisamment souple pour s'adapter aux caractéristiques locales de l'image.

L'aspect structure de la méthode présentée réside essentiellement dans la parfaite adéquation entre les caractéristiques physiques de la scène analysée et l'approximation qui est utilisée pour les représenter. Tout se passe comme si à partir d'un certain modèle du monde, on cherche à trouver la meilleure représentation d'une scène analysée avec ce modèle. Parmi toutes les configurations possibles de segmentation ayant un nombre donné de régions, il s'agit de définir comment obtenir la meilleure correspondance entre les données images et l'approximation, en jouant sur les paramètres de celle-ci et la forme des régions.

3) La propriété intervenant dans la définition d'une zone homogène joue un rôle très important. Pour qu'elle nous permette d'atteindre nos buts, il faut qu'elle traduise fidèlement l'impression visuelle de régularité et de continuité. Dans ce cas la segmentation ne peut aboutir qu'à la partition de l'image où les frontières des régions épousent les contours des objets.

2.1.2 Approche adaptative :

Par définition, un système ou une méthode sont adaptatifs si leur comportement peut être modifié en fonction de l'entrée.

Les données provenant de la numérisation d'images naturelles sont très inhomogènes. Par exemple, dans un portrait, les cheveux n'ont pas le même degré de détail que la joue ou le front. Une méthode conçue pour traiter le détail fin des cheveux ne donnera aucun résultat sur la joue ou le front si on ne peut modifier son comportement.

Toute méthode de traitement d'images naturelles se doit donc d'être adaptative si l'on souhaite des bonnes performances et une efficacité accrue.

Nous examinons ci-dessous comment l'adaptivité peut intervenir dans la segmentation.

Il existe deux approches possibles au problème de segmentation :

- la formation de frontières : cette approche consiste à trouver les frontières qui délimitent une région (approche directe),
- la formation de régions : il s'agit ici d'analyser des caractéristiques de certaines zones de l'image de manière à les découper ou à les rassembler en régions (approche indirecte).

L'attention est portée dans le premier cas sur la localisation des différences de propriétés alors que dans le second, on cherche à mettre l'accent sur les similitudes de propriétés. Il est possible que ces deux approches donnent des résultats comparables. Selon le but de la segmentation et le type de données images, il est plus simple d'utiliser l'une ou l'autre de ces méthodes.

Ainsi posé, le problème de la segmentation devient très proche de celui de l'extraction de contours. Pour les raisons déjà citées, nous adopterons l'approche indirecte. Dans ce cas, le problème apparaît plus dans la définition du modèle permettant de décrire des propriétés de zones de l'image en vue de sa partition

ultérieure. Il faut également choisir des critères pour évaluer la représentativité de ce modèle par rapport aux données de départ. En effet, une approximation bien adaptée à une certaine zone de l'image peut se révéler totalement insuffisante lorsqu'on considère une zone plus étendue. Cette considération nous amène tout naturellement à reprendre la notion de zone homogène qu'on appelle parfois "espace régulier". Cette idée fut utilisée par M. Kocher [27], puis reprise par J.C. Simon [34].

L'adaptation dans la segmentation peut intervenir soit au niveau des paramètres du modèle, soit sur le domaine où le modèle est utilisé. Dans le premier cas, les valeurs des paramètres sont modifiées de manière à vérifier l'homogénéité de la région analysée. Dans le deuxième cas, la forme de la région est modifiée pour effectuer le même test. Si aucun de ces tests ou de leurs combinaisons n'aboutit à un résultat positif, alors la frontière de la zone homogène (espace régulier) est atteinte. A partir d'un nombre restreint de pixels judicieusement choisis, il est possible d'utiliser un certain modèle pour caractériser une propriété d'une bien plus grande région de l'image. L'analyse de cette propriété pour des configurations similaires de pixels ne doit permettre que de renforcer (rendre plus robuste) le modèle.

2.1.3 Adaptation des paramètres du modèle :

Le modèle choisi pour caractériser une zone de l'image doit être simple. Il doit en outre définir le comportement du signal luminance sur cette zone. Nous allons admettre qu'à partir d'un modèle, il est possible de définir une approximation du signal original que nous noterons $g^{\wedge}(x,y)$. Pour pouvoir adapter $g^{\wedge}(x,y)$ aux données images sur un certain domaine D, il faut rendre $g^{\wedge}(x,y)$ dépendant d'un certain nombre de paramètres dont la valeur sera fixée par le choix d'un critère d'approximation du signal image.

En jouant sur tous les paramètres de l'approximation, on cherche à minimiser une certaine fonction d'erreur \mathbb{A} entre le signal original et son approximation sur le domaine D analysé. Cette fonction définit le critère d'approximation (voir section 2.3). En adaptant le modèle, c'est-à-dire les paramètres de la fonction $g^{\wedge}(x,y)$, aux caractéristiques du signal du domaine D, on peut extraire de bons attributs permettant de caractériser cette région de l'image. A partir d'un ensemble de primitives, par exemple, des polynômes bidimensionnels, on cherche quel jeu de paramètres de position, de pente, de courbure... permet de mieux représenter le signal image sur un domaine donné. On transpose une réalité complexe sur un nombre réduit de ces primitives, et ceci pour chaque objet ou partie d'objet de l'image.

2.1.4 Adaptation de la forme des régions : Division et Rassemblement :

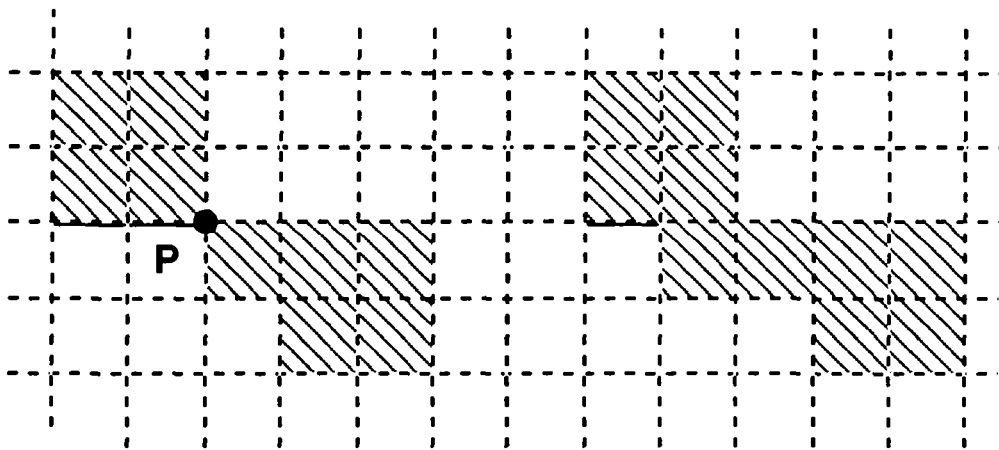
Il s'agit ici de modifier la forme et la taille de chaque domaine D de manière à garantir que \mathbb{A} ne dépasse pas un certain seuil.

Dans un premier temps, l'image est subdivisée de façon adaptative en régions de différentes tailles. Chaque région constitue une partie homogène du signal luminance que l'on est capable de reproduire presque parfaitement au moyen de notre ensemble de primitives, c'est-à-dire les paramètres de

l'approximation choisie de manière à minimiser la fonction d'erreur \mathcal{A} . Les zones ne pouvant pas être considérées comme régulières se limitent à la taille d'un pixel. Cette première phase de la segmentation appelée *division adaptative* permet ainsi d'obtenir des parties d'espaces réguliers, c'est-à-dire des parties d'objets de l'image originale. Elle fait l'objet du chapitre 3.

Supposons donc avoir obtenu cette partition initiale. Pour les régions trouvées, on est sûr que $g^{\wedge}(x,y)$ approxime fidèlement le signal image, le seuil de qualité n'ayant pas été franchi. Il s'agit maintenant de regrouper certaines régions (voire inclure des points isolés à des régions), de manière à ce que la segmentation finale corresponde aux caractéristiques physiques de la scène. Ce regroupement s'avère d'ailleurs d'autant plus nécessaire vu qu'il faut limiter autant que possible le nombre de régions pour augmenter la compression d'information. On décide ainsi de réunir deux à deux toutes les régions adjacentes qui possèdent des caractéristiques similaires.

Deux régions sont considérées comme adjacentes ou contiguës selon le principe de connexité d'ordre 4. On a préféré cette mesure de voisinage à une mesure de connexité d'ordre 8 pour deux raisons. On réduit d'une part le nombre de voisins possibles. On assure d'autre part sur le plan de l'appréciation visuelle la continuité du signal luminance sur une longueur finie (au moins la largeur du domaine d'intégration de la lumière correspondant à un pixel). En effet, en utilisant une connexité d'ordre 8, l'homogénéité de la zone peut être transmise par un point P sans dimension comme illustré à la figure 2.7.



Connexité d'ordre 8

Connexité d'ordre 4

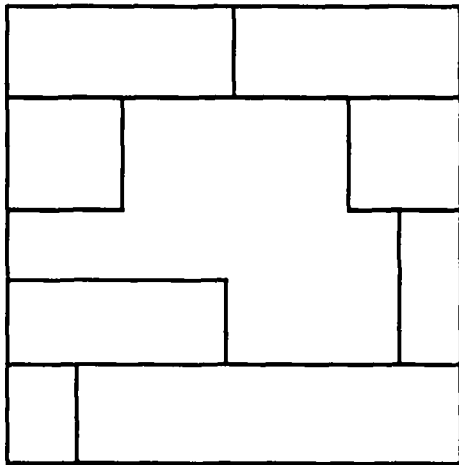
Figure 2.7: Continuité d'un domaine D connexe

- a) Connexité d'ordre 8: la continuité est assurée par un point P .
- b) Connexité d'ordre 4: la continuité est assurée sur au moins la largeur du domaine correspondant à un pixel.

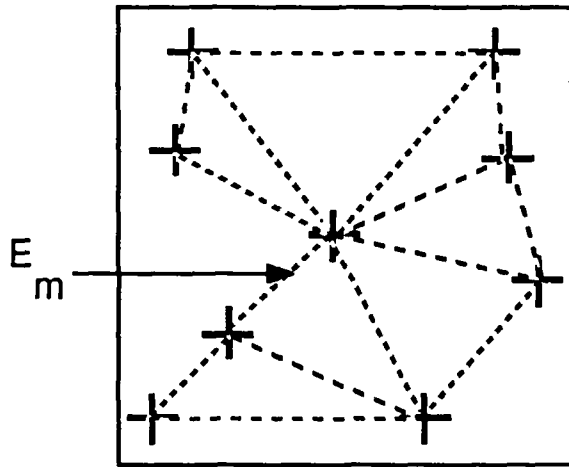
Deux domaines adjacents sont rassemblés chaque fois qu'ils correspondent à des parties d'un même espace régulier. La mesure de l'homogénéité des caractéristiques doit se faire en relation avec le changement des fonctions d'approximation $g^{\wedge}(x,y)$ sur l'ensemble des deux domaines. Moins il y a de modification, plus on peut s'attendre à ce que ces deux domaines définissent un seul et même espace régulier. Le problème reste qu'il faut placer un seuil sur la mesure d'homogénéité avant d'arrêter le processus de rassemblement. L'étude des méthodes de segmentation basées sur ce principe [27] a montré qu'au-delà de

certaines valeurs, il était illusoire de réduire le nombre de régions. On aboutit soit à une image fortement dégradée (l'information sémantique n'est pas préservée), soit à un nombre excessif de régions. On remarque en outre une forte dépendance du choix initial de la première zone à regrouper. De manière à se rendre indépendant de l'endroit où l'on commence la croissance, Kocher et al. [35] ainsi que Morris et al. [36] ont ensuite développé grâce à la théorie des graphes une technique adaptative de croissance de région.

L'idée consiste à chercher l'endroit où l'association de deux zones de l'image entraîne la plus faible dégradation. Le processus est itéré jusqu'à obtenir une segmentation jugée satisfaisante. Prenons la partition d'une image présentée à la figure 2.8a. On passe d'abord à une représentation par graphe de contiguïté de région [37]. (GCR). Chaque région constitue un noeud du graphe; chaque branche définit deux régions adjacentes. A partir du GCR, il est possible de remonter sans équivoque à l'information de voisinage de chaque région. La figure 2.8b représente le GCR associé à la figure 2.8a.



a) Partition d'une image



b) Graphe de contiguïté de région (GCR) associé

$$E_m = \text{minimum de dissimilarité}$$

Figure 2.8.

Dans un deuxième temps, on associe à chaque branche du noeud un coût qui indique le degré de dissimilarité entre les deux noeuds. En cherchant le coût le plus faible sur l'ensemble du GCR, il est possible de trouver les deux régions voisines ayant les caractéristiques les plus proches, régions que l'on peut dès lors regrouper. Après chaque rassemblement, le GCR est modifié. Les noeuds des régions où l'on a effectué le rassemblement sont associés en un seul noeud, noeud qui est relié à l'ensemble des voisins de la région ainsi constituée. Après un premier rassemblement, le second s'effectuera peut-être à un autre endroit du GCR, là où la similarité des deux régions sera jugée la plus grande. Les figures 2.9a et 2.9b donnent la partition de l'image et le GCR correspondant après une étape de regroupement en partant de la partition selon la figure 2.8a.

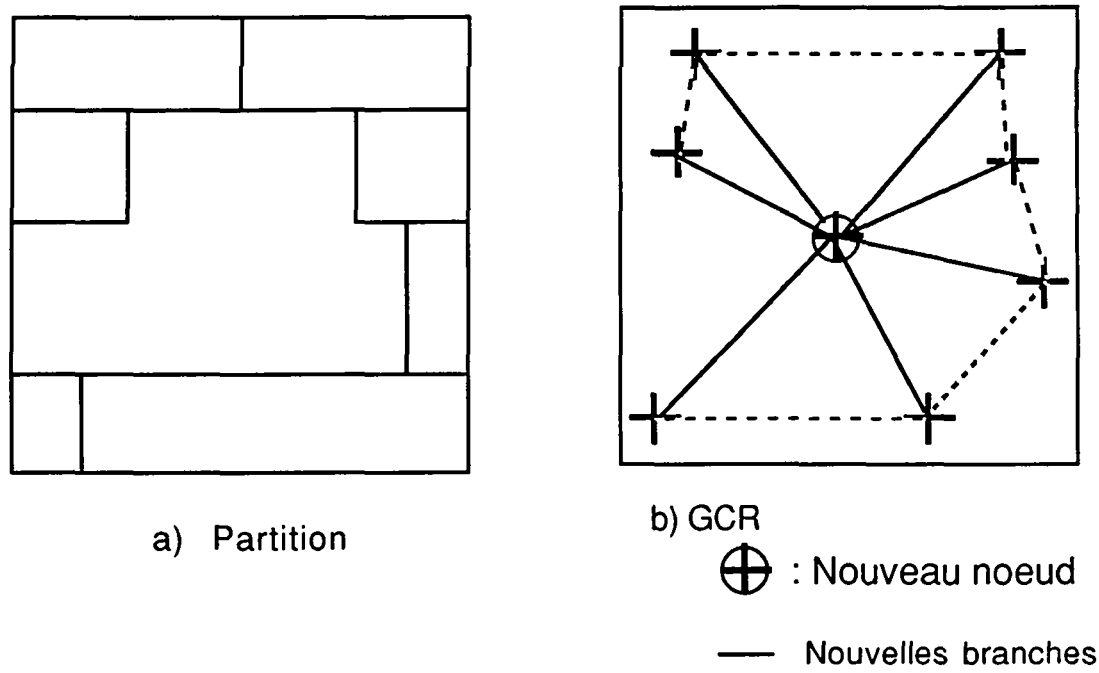
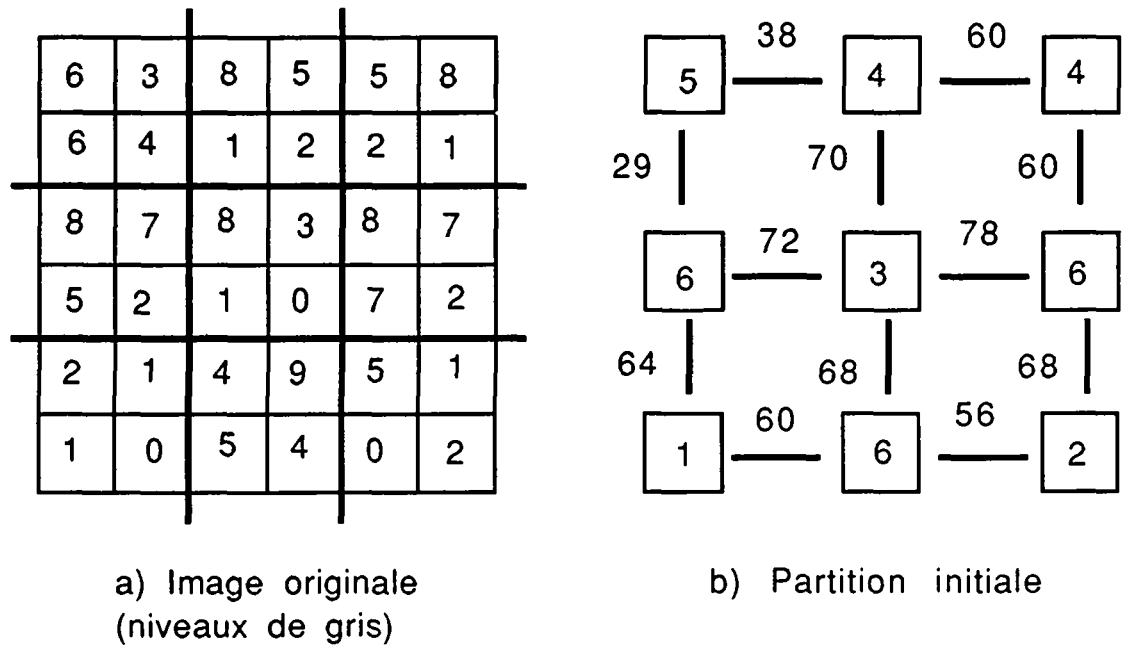
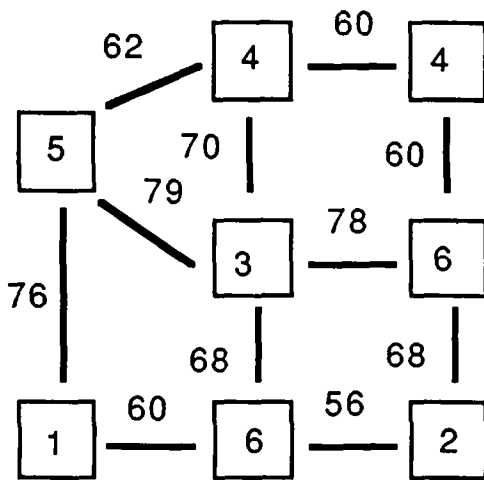


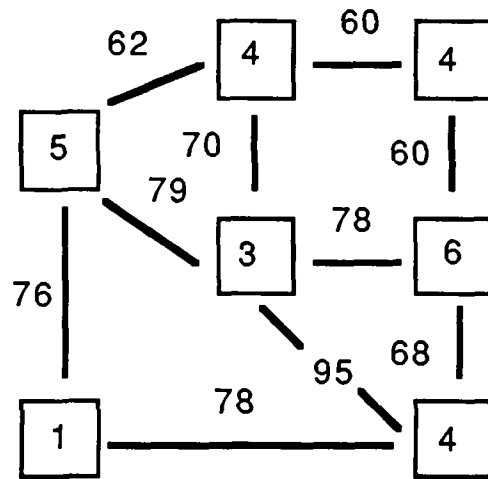
Figure 2.9: Première étape de rassemblement à partir de la segmentation de la figure 2.8.

La figure 2.10 présente deux étapes de regroupement lorsqu'on a une partition initiale de 9 carrés avec 4 pixels chacun. Pour chaque région, la dissimilarité est mesurée en estimant l'erreur quadratique si les régions sont approximées par leur valeur moyenne.





c) Partition après un rassemblement



d) Partition après deux rassemblements.

Figure 2.10: Exemple de rassemblement adaptatif. Partition initiale de 9 carrés de 2x2 pixels. A chaque noeud du GCR est associée la valeur moyenne de la région associée. A chaque branche est associée l'erreur quadratique qui résulterait après moyennage des niveaux de gris de l'association des deux noeuds que cette branche relie.

Le regroupement s'arrête lorsqu'on atteint un nombre de régions donné ou lorsqu'on commence à dégrader trop fortement l'image. Cette méthode est utilisée pour modifier la segmentation initiale (après division) par rassemblement de régions. Elle fait l'objet du chapitre 4.

La démarche de division et rassemblement est similaire à celle d'un être humain. A partir d'une appréciation globale de la scène, on descend à une analyse locale pour remonter enfin aux objets de l'image.

Une fois la segmentation obtenue, on pourra représenter l'information en décrivant la position des régions ainsi que les paramètres d'approximation judicieusement quantifiés relatifs à chacune de ces régions. A partir de ces deux informations, il sera possible de reconstruire le signal luminance approximé pour chaque région, donc chaque objet ou partie d'objet de l'image.

Si le principe de division et rassemblement que l'on vient de décrire a été proposé par Horowitz et Pavlidis [25], on l'a rendu adaptatif. On a défini aussi une manière de modéliser de façon optimale les caractéristiques du signal image, à chaque étape de la segmentation.

2.1.5 Complexité du problème :

La transformation des données images par minimisation de la fonction d'erreur A nécessite la résolution d'un système de r équations à r inconnues où r constitue le nombre de paramètres d'approximation (voir section 2.3).

Aussi bien durant la partition initiale de l'image qu'au cours du rassemblement adaptatif, il faut effectuer ce calcul chaque fois que l'on désire modifier la forme d'un domaine.

Pour chaque domaine D à regrouper à l'un de ses voisins, il faut évaluer le degré de similarité qu'il a avec chacun d'entre eux. Ensuite, l'on compare ces valeurs avec toutes celles qui sont associées aux autres branches du GCR. On a réduit la complexité du problème en limitant le nombre de régions de la segmentation avant rassemblement adaptatif par l'application de la stratégie adaptative de division. Dans [35], Kocher et al. proposent de réduire ce nombre en décomposant l'image en des carrés de même taille. Celle-ci est fixée en relation avec le nombre minimum de pixels qu'il faut considérer pour permettre leur approximation optimale au sens des moindres carrés par un polynôme bidimensionnel. Ainsi, lorsqu'on utilise un polynôme permettant de décrire une conique ($r=6$), il faut diviser l'image en un ensemble de carrés 3×3 . Cette méthode présente néanmoins deux désavantages:

- d'une part, on ne peut pas atteindre une précision extrême dans la description des frontières des régions. En effet, la méthode telle qu'elle est décrite ci-dessus limite la résolution au nombre de points se trouvant sur le côté des carrés initiaux (Dans l'exemple précédent, la précision serait réduite à 3 points). Or, ceci ne correspond en général pas à la réalité physique de l'image. A proximité de la frontière entre deux objets, il aurait fallu décomposer ultérieurement les carrés en leurs points constitutifs, ceux-ci ne pouvant définir de zone homogène.

- d'autre part, on subdivise trop l'image originale. Il est possible qu'il existe des zones beaucoup plus grandes que ces carrés, zones qui correspondent déjà à des parties d'espaces réguliers. Ces régions sont obtenues dans notre cas en imposant un seuil à ne pas franchir à la fonction d'erreur \mathbb{A} . Le rassemblement adaptatif par croissance de région selon [35] ou après division aboutit au même résultat si ce n'est qu'on a limité dans le deuxième cas le nombre de régions à rassembler.

Pour diminuer la complexité du problème sans aller changer la stratégie adaptative de rassemblement pratiquée sur l'ensemble de l'image, il suffit de réduire le nombre de paramètres d'approximation qui permettent de décrire chaque région. D'un autre côté, augmenter sensiblement le nombre de paramètres du modèle a l'avantage de limiter le nombre de régions définissant des zones homogènes, vu que l'on est à même de décrire des caractéristiques plus complexes du signal luminance.

De manière à faciliter les opérations de division et rassemblement, il apparaît essentiel d'extraire des points de repère sur l'image à traiter. C'est ce que nous nous proposons d'étudier dans ce qui suit.

2.2 Extraction d'images de contrôle :

Pour piloter le processus de segmentation, on va utiliser des images de contrôle. Celles-ci repèrent des points particuliers de l'image analysée, points correspondant à un contenu sémantique important. De ce qui a été présenté au paragraphe 2.1.1, l'information contour est primordiale dans une image. Les images de contrôle permettent de définir les contours réels des objets que certaines frontières des régions doivent épouser sans distorsion. A partir d'une image de contrôle des contours, l'erreur d'approximation entre le signal original et sa représentation $g^*(x,y)$ peut être mesurée au voisinage de ces contours. Toute valeur excessive de cette erreur entraîne une modification de la segmentation. Il est également possible de détecter l'apparition de faux contours dans l'image approximée. Ceux-ci peuvent alors être éliminés soit en changeant la segmentation, soit par l'application locale d'un filtre passe-bas de rehaussement (voir chap. 5).

2.2.1 Hypothèses sur les images traitées:

Dans ce travail, on fait l'hypothèse que les surfaces des objets que contiennent les images naturelles traitées ont une luminance à variation lente. Définir les frontières réelles des objets correspond alors simplement à extraire les contours de l'image, où ceux-ci définissent des changements brusques du signal. Pour réduire la complexité d'une scène, il faut réduire autant que possible le nombre d'objets qu'elle contient, c'est-à-dire le nombre de régions de la segmentation. Ceci peut être réalisé en utilisant des modèles plus élaborés pour décrire chaque région. Il nous a paru qu'il était plus simple de caractériser dans un premier temps des régions à variation lente qui représentent en général la majeure partie d'une image naturelle.

2.2.2 Extraction de contours:

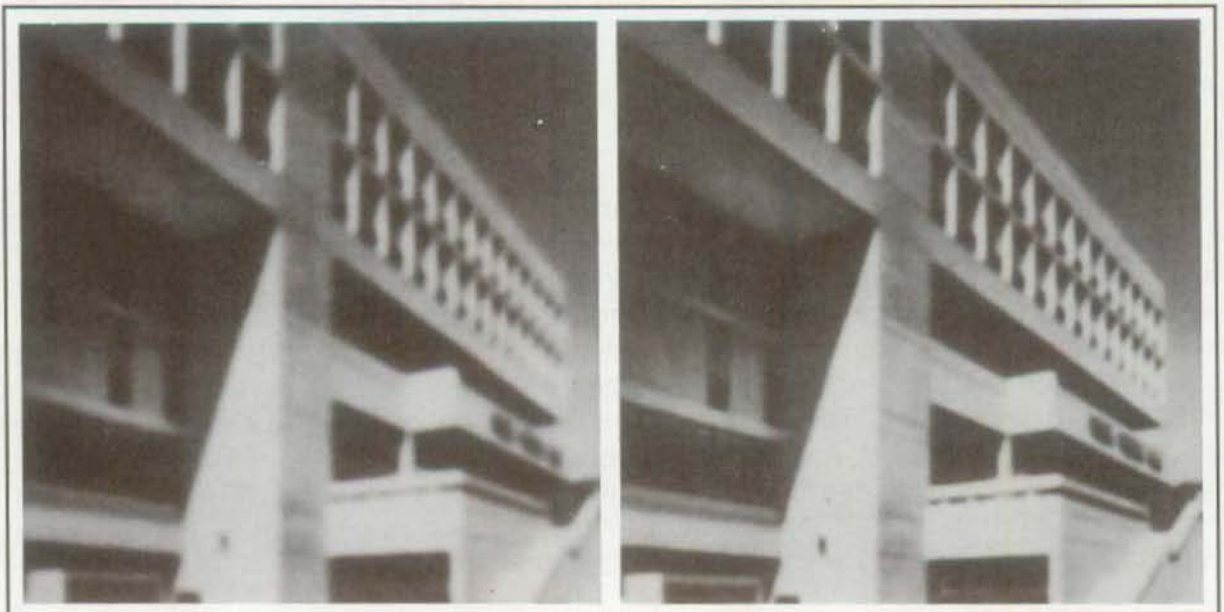
Cette étape préliminaire à la segmentation permet d'extraire une image binaire de contrôle $c(k,l)$ qui permet de localiser les contours de la scène. Ce paragraphe ne propose pas de trouver le remède miracle à ce problème qui a déjà été traité en détail mais justifie certains choix en liaison avec les buts que l'on s'est fixés. Comme on a vu que les contours se définissent par les endroits de variation rapide de luminance, on peut les détecter en estimant des dérivées du signal image. On ramène ainsi le problème d'extraction de contours à un problème de différentiation numérique. A cause de la sensibilité au bruit de la dérivation, il s'agit de régulariser l'image avant d'en calculer la dérivée [38]. Cette régularisation permet de bien poser le problème de la détection de contours au sens de Hadamard, c'est-à-dire:

- 1.- sa solution existe,
- 2.- elle est unique,
- 3.- elle dépend continuellement des données de départ.

Parmi les différentes méthodes permettant d'atteindre cette régularisation, on peut utiliser des opérateurs de stabilisation correspondant à un filtrage passe-bas.

Ceci introduit naturellement une perte de l'information haute fréquence du signal, ce qui entraîne souvent une mauvaise localisation des contours. La fonction de filtrage donnant le meilleur compromis théorique est la fonction gaussienne [38]. C'est ce qui explique en partie les performances des opérateurs de Marr-Hildreth [39] ou de Canny [40]. Dans le même ordre d'idées, Hueckel [41] a proposé d'approximer au sens des moindres carrés le signal luminance sur une région de l'image au moyen de fonctions orthogonales. Plus récemment, Haralick [42] justifia l'utilisation de fonctions polynomiales pour approximer le signal luminance de manière à augmenter la robustesse de l'extraction. D'autres techniques de régularisation peuvent être envisagées en appliquant à l'image des filtres non linéaires de lissage qui sont supposés préserver l'information contour mais éliminent la granularité locale (filtrage médian [43], filtre de Nagao [44]). Chaque fois, il est apparu que ces filtres provoquent une perte de certains détails importants tels points anguleux ou lignes contrastées. Les points anguleux correspondent aux endroits où se croisent deux contours; les lignes contrastées définissant des contours très étroits sur un fond homogène.

La figure 2.11 montre le résultat de filtrage par une gaussienne d'écart types 0.117 et 0.176 pour l'image du bâtiment et 0.059 et 0.117 pour l'image du couple dans le domaine fréquentiel (Les valeurs des écarts types sont indiquées par rapport à une fréquence normalisée à 0.5). La figure 2.12 présente l'application du filtre de Nagao après une et deux itérations pour les mêmes images de départ. Enfin, la figure 2.13 donne ces images filtrées par l'opérateur médian en utilisant des fenêtres 3x3 et 5x5. On remarque que les déformations sont souvent introduites aux points anguleux. En outre, le filtre de Nagao crèent de faux contours (visibles surtout sur le front des personnages du couple) alors que le filtrage médian tend à lisser la forme des contours (visible surtout sur le bâtiment).



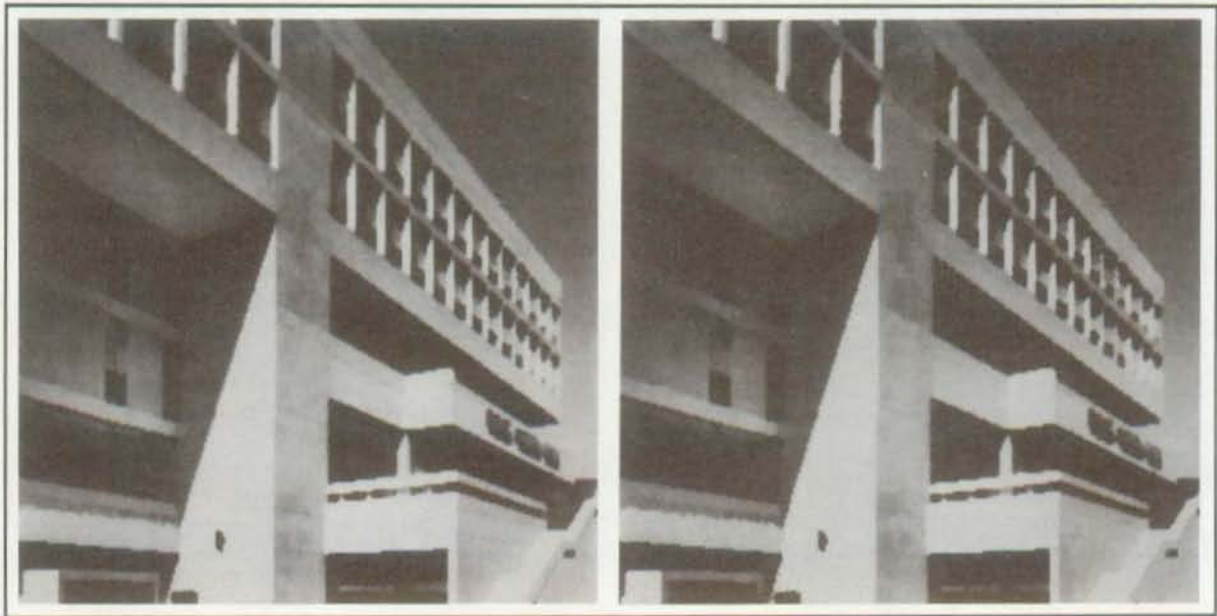
a) Bâtiment ($\sigma=0.117$)

b) Bâtiment ($\sigma=0.176$)

c) Couple ($\sigma=0.059$)d) Couple ($\sigma=0.117$)

Figure 2.11: Résultats de régularisation par divers filtres passes bas de réponse fréquentielle gaussienne

$$H_{\text{gauss}}(\rho, \theta) = \exp(-\rho^2/\sigma^2)$$



a) Bâtiment (une itération)

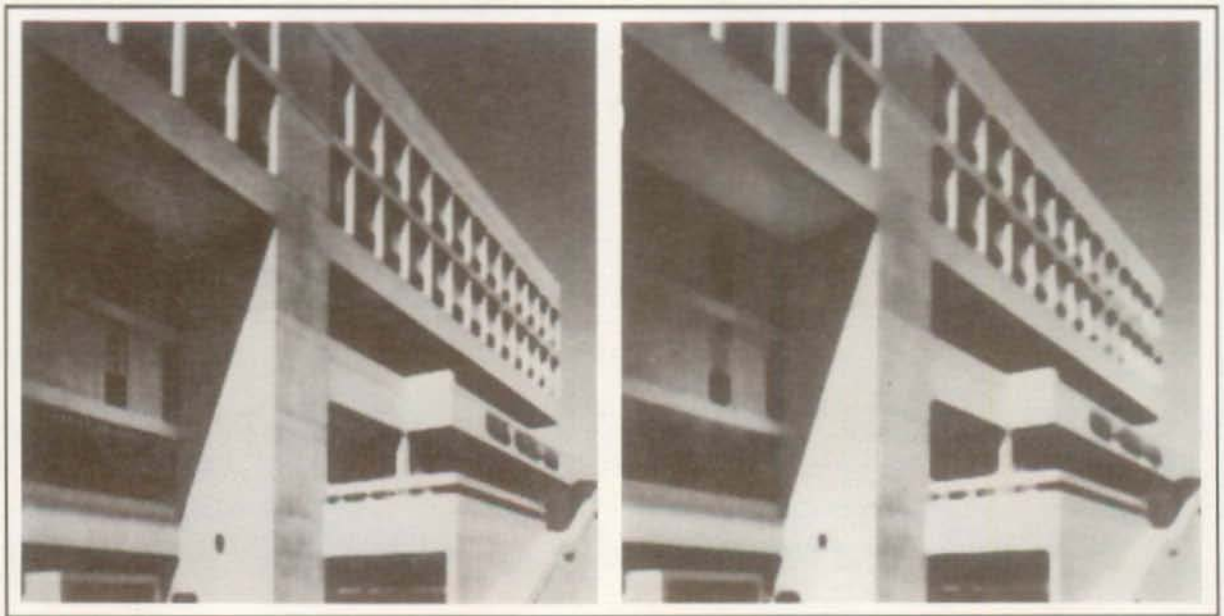
b) Bâtiment (deux itérations)



c) Couple (une itération)

d) Couple (deux itérations)

Figure 2.12: Résultats de régularisation par filtrage de Nagao 5x5 selon [44].



a) Bâtiment (fenêtre 3x3)

b) Bâtiment (fenêtre 5x5)



c) Couple (fenêtre 3x3)

d) Couple (fenêtre 5x5)

Figure 2.13: Résultats de régularisation par filtrage médian

Les contours sont détectés à partir d'une dérivée du signal régularisé. L'application d'opérateurs qui estiment la dérivée première du signal tels celui de Roberts [14] ou de Sobel [45], nécessite la recherche de maxima dans le signal filtré. Il s'agit de trouver également un moyen d'éliminer certains maxima car le signal filtré peut présenter plusieurs maxima voisins correspondant à un seul contour réel de l'image. Pour cette raison, l'utilisation d'opérateurs permettant d'estimer la dérivée seconde du signal est devenue plus courante. Les contours correspondent alors simplement aux points d'inflexion du signal filtré. Comme celui-ci est en principe à valeur moyenne nulle, ces points d'inflexion correspondront aux passages par zéro. Certains de ces opérateurs tels le Laplacien sont invariants par rotation. Il faut éviter la création de passages par zéro secondaires correspondant à des phénomènes de Gibbs. En plaçant un seuil sur l'amplitude du signal différencié, c'est-à-dire la différence entre le minimum et le maximum apparaissant de part et d'autre d'un passage par zéro, on peut éliminer bon nombre de passages par zéro secondaires. Un autre moyen consiste à élargir la largeur de la bande de transition des caractéristiques fréquentielles du filtre de différenciation. Il ne faut pas trop excéder dans ce sens sans quoi on risque d'avoir une mauvaise localisation des contours réels.

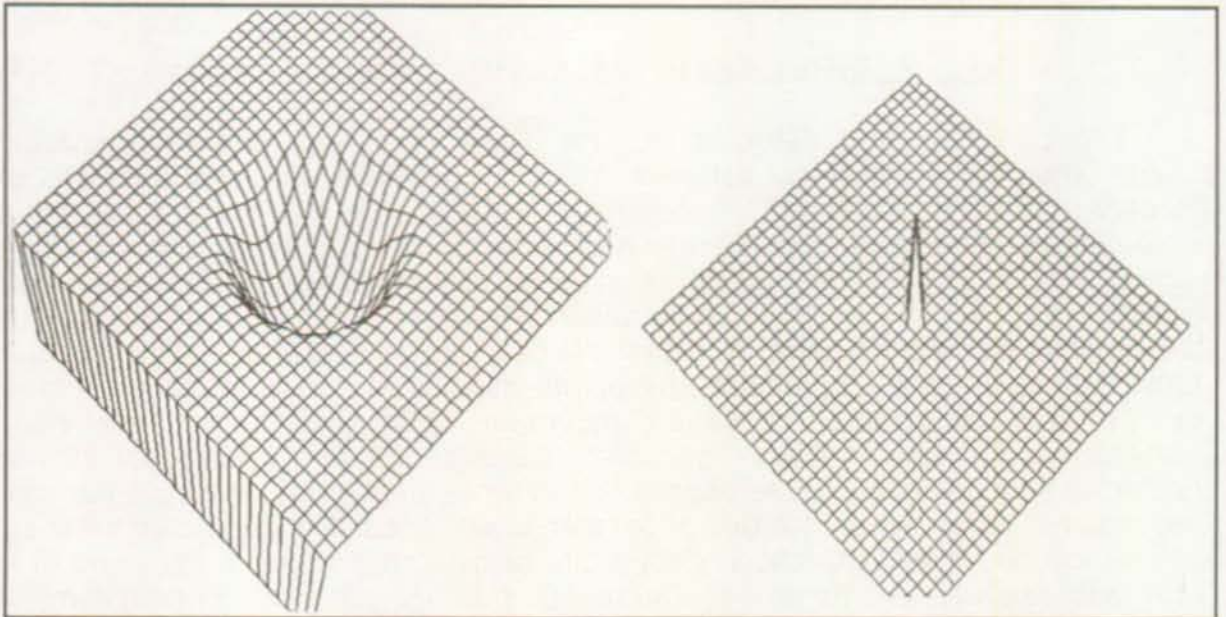
Une façon simple d'effectuer l'estimation de la dérivée seconde est obtenue à l'aide d'un filtre passe-haut idéal que l'on multiplie par une fenêtre gaussienne dans le domaine spatial pour avoir une réponse impulsionnelle finie. La convolution qui en résulte dans le domaine fréquentiel donne comme résultat une fonction erf centrée à la fréquence de coupure du filtre passe-haut idéal. Le filtre haute fréquence obtenu a donc pour réponse fréquentielle:

$$\begin{aligned}
 H(\rho, \theta) &= \{1/(\sigma\sqrt{2\pi})\} \cdot \int_{-\infty}^{\rho} \exp(-((\rho' - f_c)/(\sigma\sqrt{2}))^2) d\rho' & \text{si } \rho > 0 \\
 &= 0 & \text{si } \rho = 0
 \end{aligned} \tag{2.1}$$

où ρ et θ définissent un système de coordonnées fréquentielles polaires. Elles sont reliées aux fréquences spatiales u et v par les relations :

$$\begin{aligned} u &= \rho \cos\theta \\ v &= \rho \sin\theta \end{aligned} \quad (2.2)$$

Les paramètres σ et f_c de ce filtre permettent de s'adapter aux caractéristiques de détection de contours [8]. f_c est fixé relativement bas pour préserver la majeure partie de l'information haute-fréquence. σ doit être fixé proche de f_c pour éviter l'apparition de passages par zéro secondaires; il ne faut cependant pas lui donner une valeur supérieure à $3/4f_c$ sans quoi on risque de ne pas atténuer suffisamment les composantes très basses fréquences de l'image rendant la détection des passages par zéro erronée. Dans les exemples qui suivent $f_c=0.027$ $\sigma=0.02$ par rapport à une fréquence normalisée à 0.5. La figure 2.14 montre les réponses fréquentielle et impulsionnelle de ce filtre.



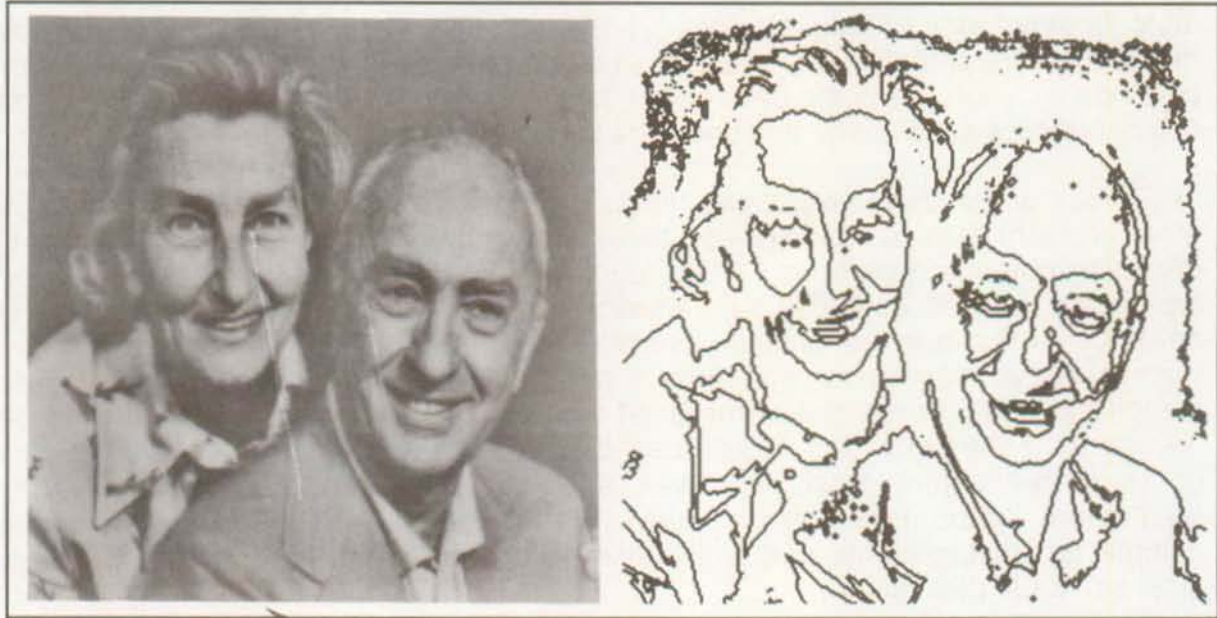
a) Réponse fréquentielle

b) Réponse impulsionnelle.

Figure 2.14: ($f_c=0.027, \sigma=0.02$)

De manière à améliorer la détection, on peut tenir compte du contexte par poursuite de contours. Les seuils sur l'amplitude des transitions aux passages par zéro sont modulés en fonction de l'existence de contours voisins déjà détectés. En partant ainsi d'un maximum de transition (contour très marqué), on recherche les chemins que définissent tous les passages par zéro liés à ce point. Ceux-ci définiront des contours de l'image si l'amplitude des transitions estimée perpendiculairement à la direction du chemin reste supérieure à un seuil de fin de poursuite. L'algorithme continue tant qu'il reste des maxima de transition dépassant un seuil de début de poursuite. Le seuil de début de poursuite est situé beaucoup plus haut que l'autre seuil, de manière à commencer toujours sur un

point de contour important. La figure 2.15 présente à cet effet l'image du couple après application du filtre de réponse fréquentielle (2.1). Plus précisément, la figure a) montre l'image filtrée, la figure b) la position des passages par zéro, la figure c) l'amplitude des transitions en ces points et enfin la figure d) les contours détectés.



a) Image filtrée

b) Position des passages par zéro de a)



c) Amplitude des transitions
aux passages par zéro

d) Contours détectés
(Seuil de début de poursuite=5.8%)
(Seuil de fin de poursuite =11.6%).

Figure 2.15: Détection des contours de l'image du couple en utilisant l'opérateur $H(\rho, \theta)$

Une question intéressante consiste à savoir s'il existe un opérateur de différentiation proche du traitement qu'effectue le système visuel humain dans la détection des contours. La physiologie de l'oeil nous apprend que certaines cellules ganglionnaires de la rétine effectuent un filtrage similaire au Laplacien d'une gaussienne, ainsi que Marr et Hildreth l'ont proposé [39]. Marr et Poggio [46] ainsi que Ikonomopoulos et Kunt [7] ont suggéré l'utilisation d'opérateurs de dérivation directionnelle en relation avec l'organisation du cortex visuel. Il est cependant difficile de comparer sur le plan psychovisuel la qualité de détection de ces opérateurs sans définir des bancs de tests exhaustifs [38].

Nous allons dans un premier temps discuter les résultats après le filtrage de Marr-Hildreth dans le contexte de notre application. La figure 2.16 présente la détection obtenue à partir de l'image du couple pour différentes valeurs de l'écart type et du seuil de détection sur l'amplitude du profil du signal à un passage par zéro. On remarque que plus le premier paramètre décroît plus on régularise l'image tout en dégradant cependant la position des contours. Un bon compromis semble possible en choisissant un écart type de 0.117 et un seuil de détection de 5% de la dynamique des amplitudes des passages par zéro. La plupart des contours réels sont ainsi préservés et il ne reste que peu de granularité. Idéalement, le seuil de détection devrait être fixé de manière locale en tenant compte de l'histogramme des amplitudes de l'image (une région contrastée aura ainsi son seuil plus élevé).



a) $\sigma=0.058$; $T=1.9\%$

b) $\sigma=0.117$; $T=1.9\%$

c) $\sigma=0.136$; $T=7.8\%$ d) $\sigma=0.117$; $T=7.8\%$

Figure 2.16: Détection des contours de l'image du couple en utilisant l'opérateur de Marr-Hildreth

$$H_{\text{Marr-Hildreth}}(\rho, \sigma) = (2/\sigma^2) \cdot [(\rho^2 / 2\sigma^2) - 1] \cdot \exp(-\rho^2 / 2\sigma^2)$$

(T : Seuil sur l'amplitude de la transition)

Si l'on se réfère aux applications de codage en relation avec les exigences définies au paragraphe 2.1.1, on voit qu'il est absolument essentiel de préserver la position de tous les contours existant dans la scène. Le filtre de Marr-Hildreth optimise le compromis détection-position mais il distord l'information des zones de l'image contenant beaucoup de contours proches (points anguleux, lignes contrastées). Ceci est dû essentiellement à l'opérateur de régularisation gaussien qui tend à avoir un effet passe-bas trop marqué sur l'image (cf. fig.2.11). Dans [39], Marr et Hildreth suggèrent de combiner les passages par zéro obtenus pour plusieurs valeurs d'écart type. Cependant aucun détail n'est donné sur la façon d'effectuer cette combinaison.

Dans le contexte des images traitées, on peut considérer qu'elles aient un rapport signal-sur-bruit élevé. Aussi, il semble plus opportun d'appliquer un simple opérateur d'estimation de dérivée seconde comme celui proposé par la relation (2.1). On a ainsi une plus grande précision bien que certains contours aient été perdus par rapport à la détection de Marr-Hildreth. Les figures 2.17 et 2.18 représentent les images du bâtiment et du caméraman filtrées avec $H(\rho, \theta)$. Les figures 2.19 et 2.20 donnent les images de contours correspondantes après poursuite comme décrit plus haut. Sauf indication contraire, ces images et celle de la figure 2.15d seront utilisées comme images de contrôle des contours dans la suite du travail.



Figure 2.17: Image du bâtiment filtrée par $H(p, \theta)$



Figure 2.18: Image du caméraman filtrée par $H(p, \theta)$.

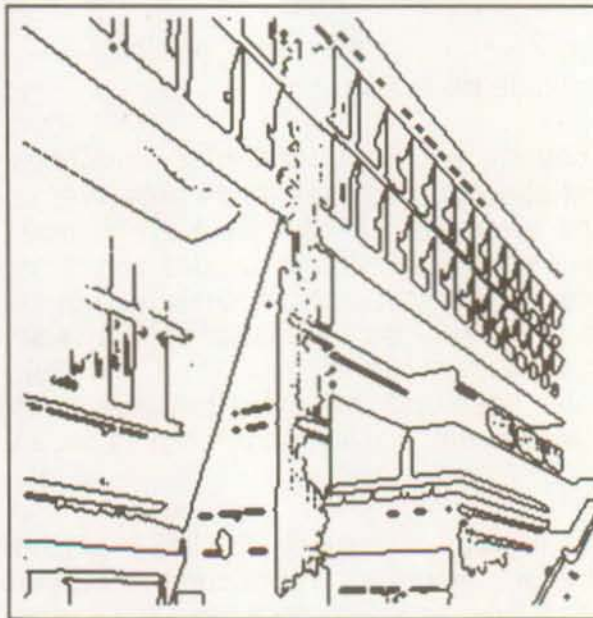


Figure 2.19: Contours détectés à partir de la fig. 2.17.

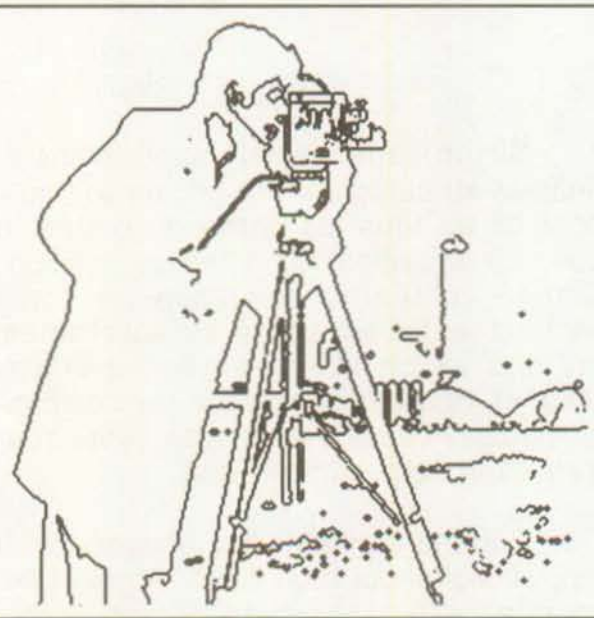


Figure 2.20: Contours détectés à partir de la fig 2.18.

Seuil de début de poursuite
Seuil de fin de poursuite

= 7.8% pour 2.19 et 19.6% pour 2.20.
= 6.6% pour 2.19 et 11.7% pour 2.20.

Les figures 2.21 et 2.22 donnent les images de contours que l'on obtient pour le bâtiment et le couple prétraités par le filtre médian 3×3 . Les résultats montrent une perte des détails comme il apparaît déjà à la figure 2.13. Des résultats similaires sont obtenus si l'on prétraite l'image avec le filtre de Nagao avant de la filtrer avec $H(p, \theta)$.



Figure 2.21: Contours détectés à partir de la fig. 2.13a.
Seuil de début de poursuite
Seuil de fin de poursuite

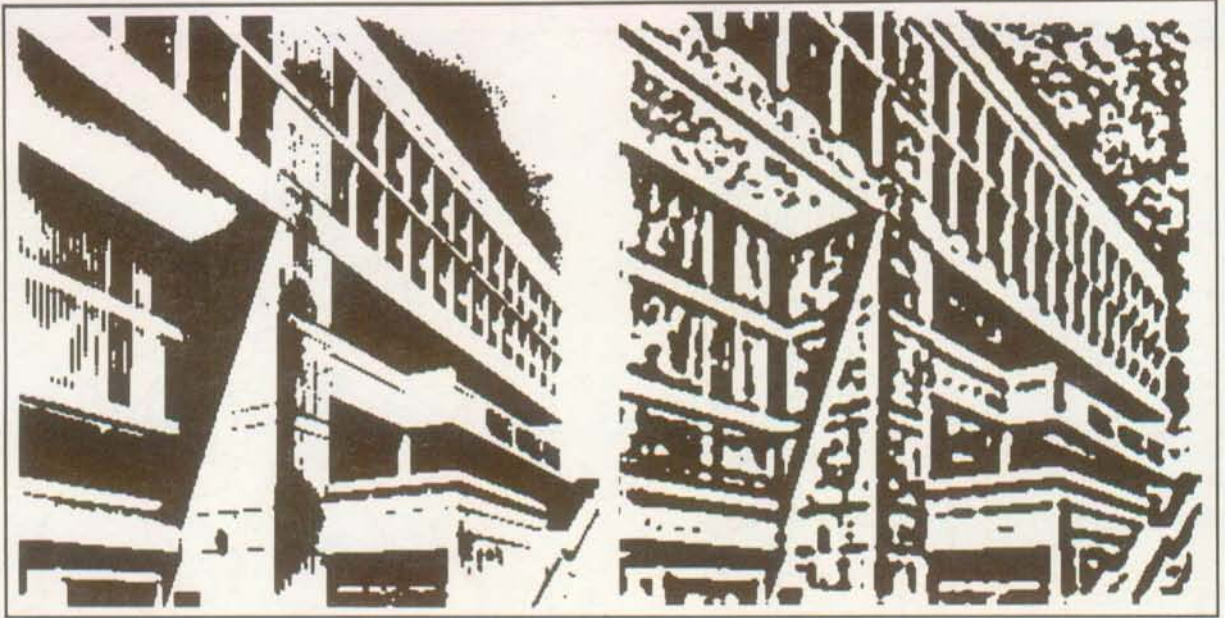
Figure 2.22: Contours détectés à partir de la fig 2.13c.
= 13.7% pour 2.21 et 19.6% pour 2.22.
= 5.9% pour 2.21 et 7.8% pour 2.22.

Si l'opération de détection de contour est déterminante pour préserver la sémantique de l'image lors de la segmentation, une autre image non moins importante peut être extraite pour regrouper certains pixels se trouvant d'un même côté d'un contour de l'image.

2.2.3 Image des structures positives et négatives :

Si l'on étudie la position des passages par zéro des images filtrées par des opérateurs de détection de contour basés sur une estimation de la dérivée seconde du signal, on remarque que les pixels situés de part et d'autre d'un tel contour et dans son voisinage, correspondent à des parties relativement homogènes du signal image. Comme il sera décrit au chapitre 4, cette partition peut se rendre utile pour faciliter le processus de rassemblement au voisinage des contours. Le nom de structures positives et négatives a été choisi en liaison avec le signe positif ou négatif de ces zones dans l'image filtrée.

Les figures 2.23a et 2.24a montrent le signe des images du bâtiment et du couple filtrées par $H(p, \theta)$. Les régions à forte activité, c'est-à-dire où le signe fluctue rapidement, ne relèvent pas d'information pertinente, les passages par zéro correspondants ne définissant pas de contours réels de l'image. Les figures 2.23b et 2.24b sont obtenues de la même façon pour les mêmes images de départ lorsqu'on a appliqué l'opérateur de Marr-Hildreth.

a) filtré par l'opérateur $H(\rho, \theta)$

b) filtré par l'opérateur de Marr-Hildreth

Figure 2.23: Image des structures positives et négatives du bâtimenta) filtré par l'opérateur $H(\rho, \theta)$

b) filtré par l'opérateur de Marr-Hildreth

Figure 2.24: Image des structures positives et négatives du couple

Une nouvelle fois, les images de contrôle obtenues à partir du filtre $H(\rho, \theta)$ seront utilisées pour piloter la segmentation, sauf autre indication.

Si l'on peut piloter la segmentation au moyen d'images de contrôle, il s'agit maintenant de définir les règles de modélisation du signal luminance pour représenter chaque zone homogène de l'image.

2.3 Approximation des données :

A la section 2.1, on a mis en évidence qu'il fallait une bonne correspondance entre l'approximation et le signal à l'intérieur de chaque région. Nous allons étudier ici le critère le plus approprié au calcul de cette approximation. Le choix d'une approximation au sens des moindres carrés va être justifié. L'analyse portera ensuite sur le genre de fonctions bidimensionnelles utilisées pour représenter le signal. Le point essentiel consiste à avoir une représentation aussi compacte que possible des données images pour chaque région, autrement dit un nombre très grand de points vont être décrits par une approximation avec peu de paramètres.

2.3.1 Type d'approximation :

La littérature abonde d'ouvrages sur la théorie des approximations [47-51]. Certains principes vont être repris ici mais il s'agit surtout de présenter les avantages de l'approximation au sens des moindres carrés dans le contexte de notre application. Les critiques généralement portées sur son utilisation pour des images naturelles proviennent du fait qu'elle a toujours été envisagée indépendamment d'une segmentation liée à la structure physique de la scène. Nous montrons que cette approximation permet une représentation efficace de la plupart des données d'une zone homogène de l'image. Les frontières de l'espace régulier sont détectées non pas par une mesure d'erreur quadratique moyenne mais par la perte ou non d'information contour. La figure 2.25 illustre ce principe.

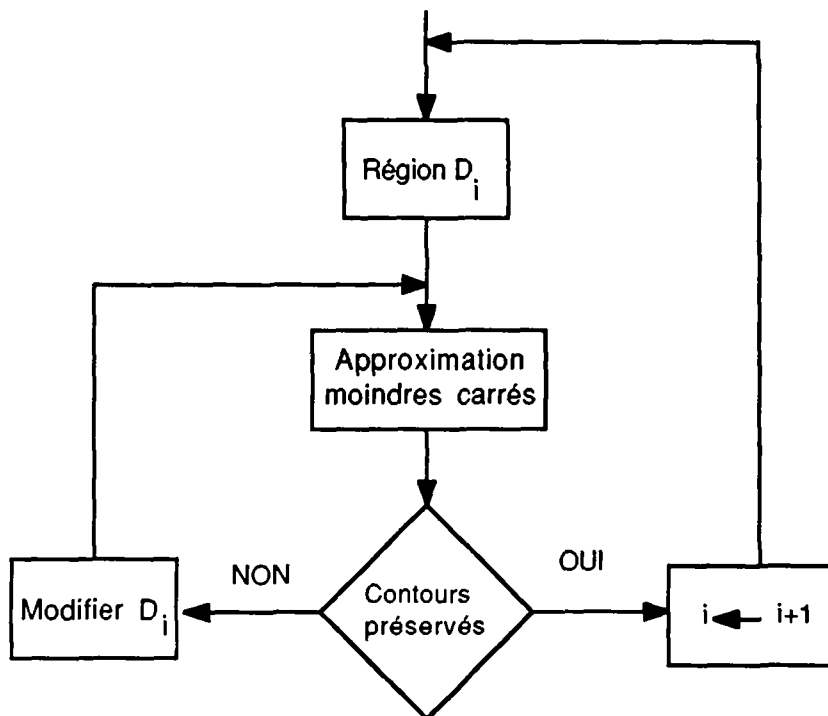
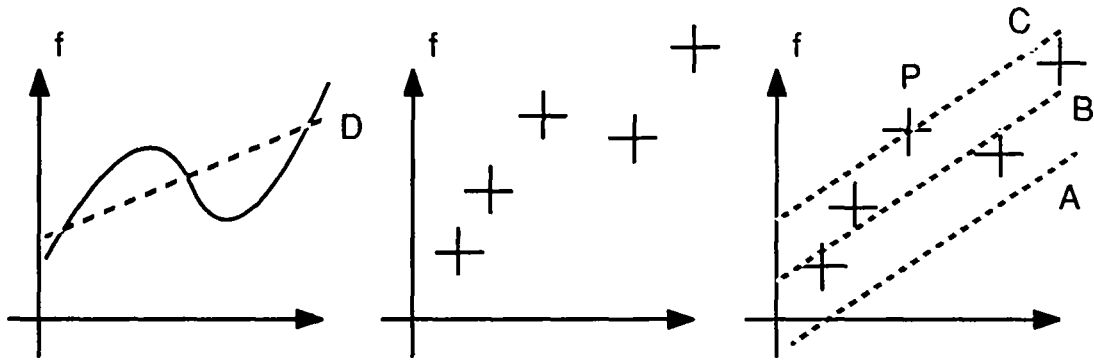


Figure 2.25: Principe de correspondance entre la segmentation et l'approximation.

Considérations générales:

Un exemple simple d'un problème d'approximation consiste à dessiner une droite qui suit au mieux une fonction continue f comme celle représentée à la figure 2.26a. Le même genre de problème peut être posé en remplaçant la fonction f par une série de points comme indiqué à la figure 2.26b. Trois choix possibles sont indiqués à la figure 2.26c. Subjectivement, on peut dire que les droites B et C sont meilleures que A. Entre B et C, tout dépend de l'importance relative que l'on veut donner au point P, d'où la nécessité d'une mesure de qualité.



a) Fonction f approximée par une droite D. b) Ensemble de points à approximer. c) Trois approximations par droites (A, B, ou C).

Figure 2.26: Exemple d'approximation d'une fonction f par une droite.

Dans cet exemple, les croix de la figure 2.26b indiquent les points à approximer. L'espace des fonctions d'approximation est défini par un ensemble de droites. Le critère d'approximation permet de choisir l'une de ces droites.

Trois considérations liées à tout problème d'approximation d'images ressortent de cet exemple :

(1) Quelle est la fonction $g(x,y)$ ou l'ensemble des données discrètes $g(k,l)$ à approximer, (k,l) correspondant à n'importe quel point du domaine bidimensionnel que l'on cherche à approximer ?

(2) Quel espace \mathbb{G} de fonctions g^{\wedge} d'approximation peut-on utiliser pour représenter ces données?

(3) Quel critère utiliser pour choisir une fonction g^{\wedge} appartenant à \mathbb{G} ?

Même si la réalité d'une image est un signal analogique, sa représentation numérique reste suffisante. Ainsi les données images peuvent être parfaitement considérées comme les données à approximer. L'utilisation d'une expression analytique $g(x,y)$ interpolant une série de pixels ne se justifie pas, cette fonction ne décrivant pas nécessairement l'évolution exacte de la luminance.

Si les données sont discrètes, les fonctions d'approximations peuvent être continues, les valeurs approximées ne rentrant en ligne de compte qu'aux lieux d'échantillonnage pour la mesure de la qualité de l'approximation. L'espace \mathbb{G} des

fonctions d'approximation est engendré par un ensemble de fonctions de base $\Psi_j(x,y)$ ($j=1, \dots, r$). Ainsi, on peut écrire :

$$g^{\wedge}(x,y) = \sum_{j=1}^r u_j \Psi_j(x,y) \quad (2.3)$$

ou sous forme vectorielle $g^{\wedge}(x,y) = \Psi^T(x,y) \cdot u$.

L'ensemble Ψ_j doit permettre d'engendrer toutes les caractéristiques possibles d'une zone homogène de l'image en jouant sur ses *paramètres* ou *coefficients* u_j .

Pour définir un critère d'approximation, il est nécessaire de définir une mesure de distance. Pour cela, considérons un espace vectoriel V muni d'une norme contenant g et g^{\wedge} , $d(a,b)$ définit la distance entre deux vecteurs a et b de cet espace si :

$$\begin{aligned} a \neq b &\Rightarrow d(a,b) > 0 \text{ et } d(a,b) = d(b,a) \\ a = b &\Rightarrow d(a,b) = 0 \end{aligned} \quad (2.4)$$

et si cette notion vérifie l'inégalité du triangle autrement dit:

$$\forall (a,b,c) \in V^3 \quad d(a,b) \leq d(a,c) + d(b,c) \quad (2.5)$$

Nous dirons que l'approximation g^{\wedge}_0 est meilleure que l'approximation g^{\wedge} du signal g si :

$$d(g^{\wedge}_0, g) < d(g^{\wedge}, g) \quad (2.6)$$

Pour tout point (k,l) du domaine D à approximer, on peut écrire :

$$g(k,l) = g^{\wedge}(k,l) + e(k,l) \quad (2.7)$$

$e(k,l)$ correspond à l'erreur commise en ce point. En général, $g(k,l)$ peut être formé d'une composante de bruit et d'une composante déterministe. Par $g^{\wedge}(k,l)$, on cherche à modéliser la composante déterministe, à caractériser ainsi l'espace régulier. Le choix des paramètres u_j va se faire de manière à minimiser une certaine fonction A des erreurs d'approximation sur tous les points du domaine D . Ceci peut être formalisé en réécrivant l'équation (2.7) de la façon suivante :

$$e(k,l) = e(k,l,u_1,u_2,\dots,u_r) = g(k,l) - g^{\wedge}(k,l,u_1,u_2,\dots,u_r) \quad (2.8)$$

où l'on met en évidence que l'erreur est une fonction des paramètres d'approximation. Chercher le minimum de A permet une transformation de l'espace des N pixels du domaine D dans l'espace des r paramètres u_j . En admettant que les pixels n'ont pas la même importance, cette fonction doit pondérer les erreurs

relatives à chacun des points. Il faut également déterminer la part respective des petites et grandes erreurs. Une expression assez générale de \mathbb{A} est donnée par :

$$\mathbb{A}(e(k,l); (k,l) \in D) = \sum_{(k,l) \in D} a(k,l) |e(k,l; u_1, u_2, \dots, u_r)|^p \quad (2.9)$$

où $a(k,l)$ et p sont des nombres réels positifs; D définit le domaine d'approximation.

Rechercher le minimum de cette fonction conduit à résoudre le système de r équations:

$$\partial \mathbb{A} / \partial u_j = 0 \quad j=1,2,\dots,r \quad (2.10)$$

$$p \sum_{(k,l) \in D} a(k,l) |e(k,l)|^{p-1} (\partial |e(k,l)| / \partial u_j) = 0$$

$$\sum_{(k,l) \in D} a(k,l) |e(k,l)|^{p-1} (\partial g^\wedge(k,l) / \partial u_j) \cdot (-\text{sgn}(e(k,l))) = 0$$

$$\text{où} \quad \text{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$$

En adoptant une notation matricielle ($\mathbf{u}^T = [u_1 \ u_2 \ \dots \ u_r]$; $\mathbf{g}^T = [g(l_1) \ g(l_2) \ \dots \ g(l_N)]$; $\mathbf{g}^\wedge{}^T = [g^\wedge(l_1) \ g^\wedge(l_2) \ \dots \ g^\wedge(l_N)]$; $\mathbf{e}^T = [e(l_1) \ e(l_2) \ \dots \ e(l_N)]$), on peut écrire les relations (2.9) et (2.10) de manière plus simple lorsque $p=2$ (approximation au sens des moindres carrés) :

$$\mathbb{A}(\mathbf{e}) = \mathbf{e}^T \mathbf{A} \mathbf{e} \quad (2.11)$$

où \mathbf{A} est la matrice diagonale $\text{diag}(a(k,l))$. l_j ($j=1,\dots,N$) définissent les N points de coordonnées (k,l) du domaine D .

La minimisation de la fonction \mathbb{A} revient à résoudre l'équation (voir Annexe A):

$$\partial \mathbb{A} / \partial \mathbf{u} = 0 \Leftrightarrow (\partial \mathbf{g}^\wedge / \partial \mathbf{u})^T \cdot \mathbf{A} \mathbf{e} = 0 \quad (2.12)$$

Si p est différent de 2, fixer les coefficients $a(k,l)$ et p de la fonction \mathbb{A} est une tâche difficile. Avant de le faire, nous allons décrire le critère d'approximation minimax ou L_∞ qui consiste à minimiser l'erreur la plus grande sur l'ensemble des données à approximer.

Critère d'approximation minimax:

Ce critère veut minimiser la mesure de distance:

$$d_{\infty}(g, g^{\wedge}) = \max_{(k,l) \in D} |g(k,l) - g^{\wedge}(k,l)| \quad (2.13)$$

Cette mesure de distance peut être exprimée au moyen de l'expression (2.9) en faisant tendre p vers l'infini (voir infra). Pour minimiser d_{∞} , on peut procéder à la démarche itérative suivante:

a) choisir $r+1$ points $l_i \in D$ que l'on notera l_1, \dots, l_{r+1} ; on va s'arranger pour avoir la même valeur absolue e de l'erreur d'approximation en chacun de ces points,

b) résoudre le système de $r+1$ équations $g(l_i) = (-1)^i e + \sum_{j=1..r} u_j \Psi_j(l_i)$; l'erreur e et les coefficients u_j définissent les $r+1$ inconnues de ce système.

c) calculer, avec les paramètres u_j solution du système précédent, l'erreur aux points l_{r+2} à l_N restant du domaine D ,

d) chercher le maximum de ces erreurs E ,

e) si $E=e$, le processus s'arrête; sinon recommencer à partir de a) en remplaçant un des points l_1 à l_{r+1} par le point représentant l'erreur E .

On remarque que la solution optimum donne la même erreur e en chaque point du domaine à approximer. Ce critère est donc très sensible au bruit. Comme chaque région définissant une zone homogène contient nécessairement du bruit, ce genre d'approximation introduit une distorsion en tout point même où le bruit est presque inexistant. On n'arrive donc pas à reproduire les caractéristiques moyennes de la région. En outre, si l'on doit encore effectuer la partition de l'image, il devient très difficile de savoir comment modifier la région, l'erreur d'approximation e étant identique en chaque point. Ainsi, si l'on désire modifier un certain domaine, il faut pratiquement analyser parmi toutes les possibilités de modification celle qui produit la plus faible erreur d'approximation selon ce même critère L_{∞} . Enfin, la présence de contours à l'intérieur d'une région entraîne le plus souvent une approximation minimax présentant de faux contours, à cause de l'importante valeur de e de la solution optimale. En ce qui concerne le temps calcul, cette solution est très coûteuse (plusieurs inversions de matrices $(r+1) \times (r+1)$ selon le nombre d'itérations nécessaires), rendant sa mise en oeuvre pratique irréalisable.

Ce critère d'approximation s'avère donc inefficace pour modéliser les données images à l'intérieur d'un domaine D . Il s'agit alors d'étudier les effets des paramètres $a(k,l)$ et p de l'expression (2.9). Pour simplifier, appelons e_i les erreurs $e(l_i)$ en chaque point du domaine D .

Effet du paramètre p :

Si sa valeur passe de p_1 à p_2 avec p_2 plus grand que p_1 , la minimisation de \mathbb{A} entraîne une diminution des plus grandes valeurs e_i obtenues avec une solution optimale utilisant p_1 . Ceci correspond d'un autre côté à une augmentation des autres erreurs, de telle sorte que la dynamique des erreurs possibles tend à se réduire au fur et à mesure que p devient grand. A la limite, lorsque p tend vers

l'infini, on retrouve la solution minimax qui donne une erreur identique en chaque point. De la même manière, l'approximation optimale pour de faibles valeurs de p tend à garder la plupart des erreurs faibles. Si une région ne contient pas de contours, on est ainsi assuré d'avoir une bonne correspondance entre le signal original et son approximation à l'intérieur du domaine analysé. Par contre, en présence de contours, il n'y a correspondance que si l'on a à faire à de petites régions sans quoi, il faut modifier la forme de la région considérée jusqu'à faire épouser l'une de ses frontières avec le contour en question. Si de faibles valeurs de p semblent ainsi bien adaptées, la résolution du système (2.10) est fortement simplifiée en choisissant p égal à 2. En effet, il suffit ainsi de résoudre le système d'équations (2.12).

Effet des coefficients de pondération $a(k,l)$:

Vu que l'information contour est de première importance, il semble nécessaire de forcer l'approximation à suivre le signal original au voisinage des contours. Bien entendu, il s'agit plus de préserver la position du saut de luminance que son amplitude. C'est pourquoi, si une certaine région contient un contour il s'agira de pondérer fortement les erreurs en fonction du gradient du signal original. Néanmoins, ceci ne doit pas se faire au détriment d'une certaine fidélité dans la reproduction du signal à variation lente de part et d'autre du contour. On risquerait en outre de créer des contours artificiels, violant ainsi les exigences de départ. Ces artefacts sont fortement fonction de l'espace des configurations qui peut être engendré par la modification des coefficients d'approximation u_j . En utilisant des fonctions d'approximations polynomiales, on a pu relever que dans le contexte d'une grande région contenant un contour:

1. toute pondération fonction du gradient tend à détruire l'information de part et d'autre du contour pour des polynômes de degré inférieur ou égal à 4 (conjecture).

2. une représentation du saut de luminance est impossible si aucune pondération n'est effectuée. Dans un tel cas de non pondération, on remarque une erreur importante au voisinage du contour (conjecture).

La première conjecture justifie donc le choix d'éviter de pondérer les erreurs d'approximation selon leur position dans la région considérée. Ceci présente d'autant plus l'avantage de simplifier les calculs. Le contenu de la deuxième conjecture ne peut être évité qu'en modifiant la forme de la région de manière à faire épouser certaines de ces frontières avec le contour. Les mesures d'erreur le long de ces contours permettent de changer la partition de l'image comme on le verra à la section suivante.

Approximation au sens des moindres carrés:

Le critère d'approximation retenu correspond donc à minimiser la somme des erreurs quadratiques. La solution optimale peut être obtenue en reprenant les relations (2.10) et (2.12), par la résolution du système

$$\mathbf{z}^T (\mathbf{g} - \mathbf{Z} \mathbf{u}) = 0 \quad (2.14)$$

où \mathbf{Z}^T est la matrice $r \times N$ $[\Psi(l_1) \Psi(l_2) \dots \Psi(l_N)]$.

En effet, le vecteur des valeurs approximées peut s'exprimer par $\mathbf{g}^\wedge = \mathbf{Z}\mathbf{u}$. La relation (2.14) donne comme vecteur optimal de coefficients:

$$\mathbf{u} = (\mathbf{Z}^T \mathbf{Z})^{-1} \cdot \mathbf{Z}^T \mathbf{g} \quad (2.15)$$

Pour simplifier, nous noterons \mathbf{S} la matrice symétrique définie positive $r \times r$ $\mathbf{Z}^T \mathbf{Z}$ et \mathbf{h} le vecteur $\mathbf{Z}^T \mathbf{g}$. L'équation (2.15) ne peut être résolue que si \mathbf{S} est non singulière. L'élément général de cette matrice s'écrit:

$$s_{pq} = \sum_{(k,l) \in D} \Psi_p(k,l) \cdot \Psi_q(k,l) \quad (2.16)$$

Lorsque une solution unique au problème existe, la résolution de (2.15) requiert:

- 1.- l'évaluation des vecteurs $\Psi(k,l)$ pour tous les points du domaine D ,
- 2.- $Nr(r+1)/2$ multiplications et $(N-1)r(r+1)/2$ additions pour calculer la matrice \mathbf{S} . Pour simplifier, on considèrera que $O(Nr^2)$ opérations suffisent,
- 3.- La multiplication matrice-vecteur pour estimer le vecteur \mathbf{h} ce qui implique $O(Nr)$ opérations,
- 4.- la résolution du système $\mathbf{S} \cdot \mathbf{u} = \mathbf{h}$, ce qui correspond en général à $O(r^3)$ opérations [52].

On voit donc que les étapes 1 à 3 sont les plus coûteuses comme en général $N \gg r$. Pour pouvoir réaliser la modification successive de la segmentation en évaluant chaque fois la meilleure approximation au sens des moindres carrés, on étudiera aux chapitres suivants comment il est possible d'optimiser ce traitement, en tenant compte de diverses propriétés de ce genre d'approximation.

2.3.2 Fonctions d'approximation:

Le choix des fonctions Ψ_j est fondamental. En effet, c'est elles qui permettent de générer l'espace des approximations possibles en jouant sur les coefficients u_j .

Le choix d'un ensemble complet de fonctions orthogonales ne se justifie pas car la méthode proposée implique la modification systématique de la forme des régions. Ceci entraîne une évolution de l'ensemble des fonctions orthogonales sur chaque région. Si le calcul de l'approximation est simplifié, la matrice \mathbf{S} pouvant se ramener à l'identité après normalisation, une procédure d'orthogonalisation [53] de l'ensemble des fonctions devient nécessaire. En outre, aucune considération a priori ne permet de choisir un sous-ensemble r de ces fonctions pour calculer l'approximation. Il faudrait estimer la transformation de l'espace image dans l'espace des N fonctions orthogonales sur le domaine D et garder les r coefficients

les plus importants. Au niveau d'un système de codage, il devient alors nécessaire de transmettre pour chaque région les r fonctions correspondantes retenues.

A partir d'un ensemble de r fonctions de base Ψ_j , rien ne sert de les orthogonaliser sur le domaine D où l'on désire calculer l'approximation. En effet, cette transformation correspond à un simple changement de base, si bien que la fonction approximée g^* possède exactement la même expression sur D . Ceci est démontré dans l'annexe B. Les résultats de la méthode de segmentation sont donc identiques à ceux obtenus sans orthogonalisation de l'ensemble des fonctions de base, si ce n'est que les calculs sont compliqués davantage.

Comme on suppose que dans la segmentation finale, il n'existe pas de régions contenant des contours, il est raisonnable de dire que le signal à l'intérieur de ces régions est à variation lente, selon les hypothèses formulées au paragraphe 2.2.1.

Les fonctions polynomiales sont bien adaptées à représenter de telles variations. En outre, la régularité des surfaces engendrées par des polynômes est grande: cela évite d'avoir de phénomènes d'oscillations désagréables sur le plan visuel, que l'on aurait en utilisant des fonctions orthogonales périodiques (Fourier, Hadamard,...). Si une bonne correspondance est assurée sur une grande partie du domaine d'approximation, les fonctions polynomiales présentent l'avantage de s'éloigner rapidement aux limites de ce domaine, garantissant la non destruction de contours existants.

Nous admettrons donc de représenter les différentes régions au moyen de polynômes bidimensionnels. On peut écrire généralement:

$$\Psi_j(x,y) = x^p \cdot y^q \quad (2.17)$$

Le nombre de ces fonctions reste faible de manière à "transformer" beaucoup de pixels en peu de coefficients d'approximation. On garantit ainsi une bonne simplification de la scène ainsi qu'une compression plus élevée pour un nombre donné de régions. En maintenant r relativement bas, on réduit également la complexité de calcul vu que celle-ci croît avec r . Les monômes de base exprimés par la relation (2.17) ne doivent pas être de degré trop élevé pour éviter des erreurs de troncation dans le calcul de l'approximation. Pour pouvoir décrire une grande variété de surfaces à variation lente, on utilisera des fonctions polynomiales de degrés 0 à 3 impliquant respectivement 1,3,6 et 10 coefficients. Plus précisément, on peut écrire :

$$\begin{aligned} P_0(x,y) &= u_1 \\ P_1(x,y) &= u_1 + u_2x + u_3y \\ P_2(x,y) &= u_1 + u_2x + u_3y + u_4x^2 + u_5xy + u_6y^2 \\ P_3(x,y) &= u_1 + u_2x + u_3y + u_4x^2 + u_5xy + u_6y^2 + u_7x^3 + u_8x^2y + u_9xy^2 + u_{10}y^3 \end{aligned} \quad (2.18)$$

Nous verrons au chapitre suivant comment il peut être intéressant d'avoir des fonctions polynomiales entraînant une formulation séparable du problème.

2.3.3 Quantification de l'approximation:

Une fois les coefficients d'approximations optimaux calculés, on estime l'approximation g^{\wedge} en chaque point de la région considérée. Si la dynamique du signal original à 8 bits est fixée entre 0 et 255, il s'agit d'y inclure également celle de l'approximation g^{\wedge} . Si la dynamique de g^{\wedge} dépasse celle du signal original, une première possibilité consiste à requantifier linéairement le signal approximé en 256 niveaux. Néanmoins, on introduit ainsi une erreur en chaque point. Il semble donc plus judicieux de tronquer les points se trouvant au-delà des seuils de luminance de manière à les ramener à ces valeurs. On arrondit les valeurs des autres points aux entiers les plus proches. Si la dynamique du signal approximé s'étend de α à β avec $\alpha \leq 0$ et $\beta \geq 255$,

- par quantification linéaire entre 0 et 255, on commet en chaque point une erreur:

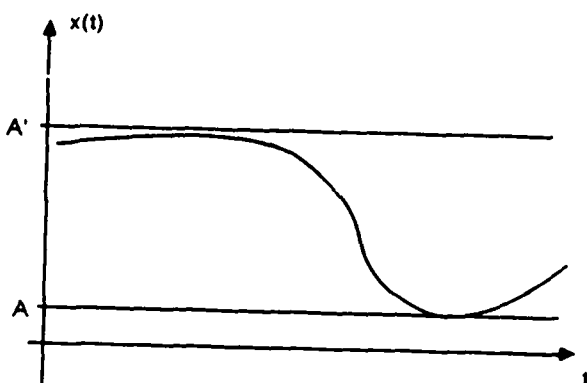
$$e_q = |g - 255.(g^{\wedge} + \alpha)/(\beta - \alpha)| \quad (2.19)$$

au lieu de $|g - g^{\wedge}|$. On risque au plus d'augmenter l'erreur de

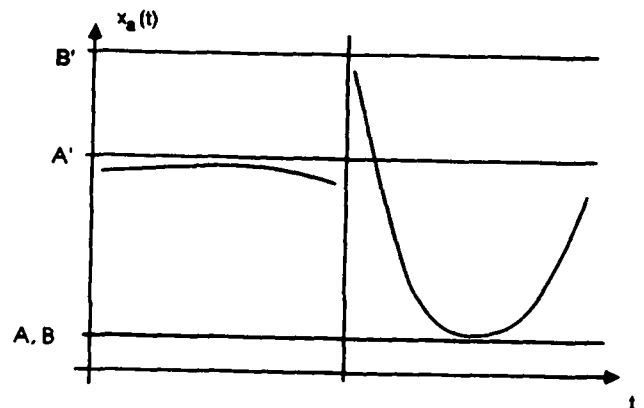
$$|255.(g^{\wedge} + \alpha)/(\beta - \alpha) - g^{\wedge}| + 0.5 = |g^{\wedge}(1 - 255/(\beta - \alpha)) + 255\alpha/(\beta - \alpha)| + 0.5 \quad (2.20)$$

Le terme 0.5 provient de l'arrondissement de la valeur absolue à l'entier le plus proche.

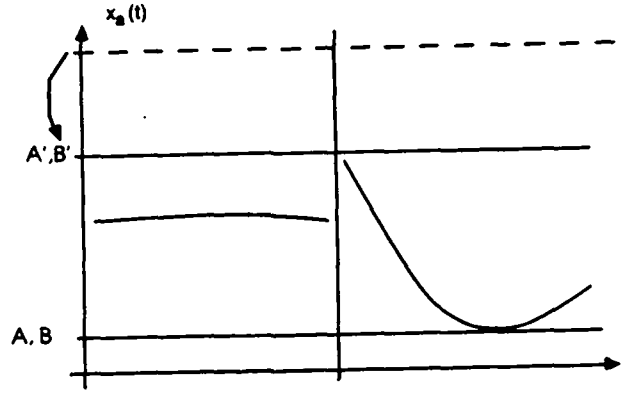
- par troncation des valeurs sortant de l'intervalle $[0, 255]$, on augmente l'erreur en chaque point compris dans cette gamme au plus de 0.5 alors qu'elle est réduite de α pour les valeurs en-dessous de 0 et de $\beta - 255$ pour celles au-dessus de 255. Il y a diminution de ces valeurs car le signal de départ était compris entre 0 et 255. Cette deuxième solution sera donc retenue. La figure 2.27 illustre par ailleurs les problèmes de perte de contraste qui peuvent résulter du choix de la quantification linéaire.



a



b



c

Figure 2.27: Illustration de la perte de contraste résultant d'une requantification linéaire.

En 2.27a, on présente un signal original et sa gamme dynamique AA'. En 2.27b, on a une représentation de l'approximation du signal original sur deux régions au moyen de deux arcs de parabole; la dynamique correspondante s'étend sur le segment BB'. Enfin, en 2.27c apparaît la perte de contraste résultant de la requantification linéaire de BB' à AA'.

Le problème de l'approximation ayant été résolu, il s'agit maintenant de définir les critères nécessaires à estimer la qualité de la segmentation pour en justifier une éventuelle modification.

2.4 Mesures de qualité :

Deux indices de qualité seront mis en évidence dans ce qui suit. Le premier est basé sur des estimations globales telles erreur quadratique alors que le second définit une mesure plus contextuelle basée sur la bonne représentation de l'information contour. Une stratégie évolutive doit être mise sur pied pour tenir compte de la structure des régions dans le choix des mesures d'erreur.

2.4.1 Mesures globales:

Considérons pour ce faire l'erreur quadratique (EQ) sur le domaine D analysé. Celle-ci s'exprime par:

$$\begin{aligned} EQ &= (\mathbf{g} - \mathbf{Z} \mathbf{u})^T (\mathbf{g} - \mathbf{Z} \mathbf{u}) \\ &= \mathbf{g}^T \mathbf{g} - 2 \mathbf{u}^T \mathbf{Z}^T \mathbf{g} + \mathbf{u}^T \mathbf{Z}^T \mathbf{Z} \mathbf{u} \end{aligned} \quad (2.21)$$

où \mathbf{u} représente le solution optimale de l'approximation au sens des moindres carrés sur D. En utilisant (2.15), on voit que $\mathbf{Z}^T \mathbf{Z} \mathbf{u} = \mathbf{S} \mathbf{u} = \mathbf{Z}^T \mathbf{g} = \mathbf{h}$. La relation (2.19) se ramène dès lors à :

$$EQ = \mathbf{g}^T \mathbf{g} - \mathbf{u}^T \mathbf{h} \quad (2.22)$$

Cette mesure de qualité ne donne qu'une estimation globale de la représentativité de l'approximation sur D. De manière à ne pas perdre l'information contour, il faut rester sévère quant au seuil à ne pas dépasser. Comme le montreront les résultats de découpage présentés au chapitre suivant, ceci entraînera une segmentation de l'image en un nombre élevé de petites régions.

On peut diminuer ce nombre en rendant ce critère de qualité relativement indépendant de la taille de la région. Il suffit de pondérer l'erreur quadratique par les N points du domaine aboutissant ainsi à ce qui est appelé l'erreur quadratique moyenne (EQM). On a donc la relation $EQM = EQ/N$. Par rapport à EQ, la segmentation donne une partition de l'image avec des régions de plus grande taille, mais il faut toujours placer le seuil relativement bas de manière à préserver les contours. Il semble donc nécessaire d'introduire des mesures de qualité plus contextuelles qui tiendront compte de la présence de contours dans le domaine analysé. Par opposition aux critères considérés jusqu'ici, ces mesures seront évaluées localement.

2.4.2. Mesures contextuelles:

Pour mesurer la qualité de reproduction des contours, il faut dans un premier temps utiliser l'image de contrôle des contours pour déterminer l'emplacement de ceux-ci. Ensuite, une erreur de contour (EC) est estimée. On évalue pour cela l'erreur commise au niveau des pixels se trouvant de part et d'autre du contour. De manière à rendre cette mesure indépendante de la longueur des contours apparaissant dans l'image on pondère les erreurs par le nombre de points impliqués par cette mesure. On peut ainsi écrire:

$$EC = (1/N_C) \cdot \sum_{(k,l) \in (D \cap C)} e^2(k,l) = (1/N_C) \sum_{(k,l) \in (D \cap C)} (g(k,l) - g^{\wedge}(k,l))^2 \quad (2.23)$$

où C représente l'ensemble des points situés de part et d'autre de tous les contours de l'image. N_C définit le nombre de points de l'intersection $(D \cap C)$. Bien entendu, l'efficacité de cette mesure dépend de l'exactitude de l'image de contrôle des contours.

Une autre possibilité consiste à pondérer les erreurs au voisinage des points contours par une estimation du gradient de l'image en ces points. Dans ce cas, la dégradation peut être calculée en partant de la formule (2.10) avec $a(k,l)$ correspondant à la composante du gradient normale au contour.

Si EQ ou EQM permettent de voir la correspondance moyenne qui existe entre le signal original et son approximation, EC détermine si des contours réels sont perdus. Ces critères sont élaborés pour pouvoir mesurer la qualité de la segmentation et non celle de l'approximation.

Définir un seuil de qualité acceptable est une tâche difficile si l'on n'a pas d'idée a priori de ce qu'on est en train d'analyser. La technique adaptative de rassemblement permet d'éviter ceci car on regroupe à chaque fois les deux régions présentant le moins de dissimilarité. Il reste à définir cette dissimilarité.

Si la présence de contours justifie la définition de mesures appropriées, il en va de même des mesures en liaison avec la forme et la taille des régions. En effet, selon les régions impliquées dans le processus de rassemblement, il apparaît nécessaire de faire évoluer les critères permettant d'estimer la ressemblance entre ces régions. La section 4.5. définira la stratégie à adopter en liaison avec la structure des régions. Il faudra auparavant analyser l'effet des différentes mesures de dissimilarité envisagées.

Commentaires:

1) Le premier terme de (2.22) correspond simplement à la somme des carrés des valeurs de l'image sur le domaine D . C'est donc l'énergie de ce signal sur ce domaine. Le second $\mathbf{u}^T \mathbf{h} = \mathbf{g}^T \mathbf{g}$ définit quant à lui l'énergie du signal d'approximation. Ainsi, si l'on adopte le critère des moindres carrés, l'erreur quadratique est exprimée par la différence d'énergie du signal original et de son approximation.

2) Le calcul de l'énergie du signal d'approximation nécessite uniquement $O(r)$ opérations, les vecteurs \mathbf{u} et \mathbf{h} ayant été estimés lors du calcul de l'approximation. On évite ainsi le calcul point par point du signal approximé sur D , qui entraînerait $O(Nr)$ opérations supplémentaires.

3) Le calcul de (2.23) requiert l'évaluation de N_c pixels approximés, ce qui peut être réalisé en $O(N_c r)$ opérations.

2.5 Conclusion :

Ce chapitre a souligné l'importance d'une approche adaptative pour segmenter une image. En effet, la nécessité de préserver l'information contour en recherchant les espaces réguliers de l'image ne peut se faire qu'en adaptant itérativement la forme des régions et de l'approximation en liaison avec les données originales. De manière à limiter la complexité d'analyse, on a proposé une stratégie de division et rassemblement dont les détails vont être présentés dans ce qui suit.

Un prétraitement a été mis en oeuvre de manière à extraire des images de contrôle des contours servant à piloter les différentes phases de la segmentation. Le choix du critère des moindres carrés pour approximer les données a été justifié dans le contexte de la segmentation d'images. Le signal luminance de chaque région est reconstruit en utilisant des polynômes bidimensionnels en raison de la correspondance qui existe entre des portions de scènes naturelles et ce genre de fonctions. Enfin, ce chapitre a servi à mettre en évidence l'utilisation de mesures globales et contextuelles pour estimer la qualité de la segmentation. Les premières donnent la correspondance moyenne qui existe entre le signal original et son approximation à l'intérieur d'une région. Les secondes garantissent de préserver les contours de l'image et l'élaboration correcte de la procédure de rassemblement en liaison avec la structure des régions.

3. Opération de division

Ce chapitre présente la première phase de la segmentation : la division ou le découpage. L'image de départ est *subdivisée de manière adaptative en une série de carrés* de différentes tailles qui seront ensuite regroupés lors du processus de rassemblement.

Chaque carré définit une partie d'un espace régulier de l'image. Il s'agit donc de garantir tout au long du processus de division une bonne correspondance entre les données originales et l'approximation à l'intérieur de chacun de ces carrés. Pour ce faire, on estimera la qualité de l'approximation au moyen de l'erreur quadratique moyenne (mesure globale) et de l'erreur le long des contours présents dans le carré (mesure contextuelle).

Le chapitre est organisé en six sections. Dans un premier temps, on définit les critères de subdivision de l'image originale. On s'attache ensuite à décrire comment structurer les données pour implanter l'algorithme de division. La troisième section présente la complexité de calcul du problème et montre comment il est possible d'optimiser les différentes phases du traitement. La quatrième section décrit comment représenter l'information après division. Dans la cinquième section, on analyse expérimentalement les effets des différents paramètres de l'algorithme en appliquant celui-ci à plusieurs images naturelles. Cette partie du chapitre donne un aperçu de la compression de l'information qui est déjà réalisée à ce niveau du traitement. La dernière section résume l'exposé qui précède.

3.1 Principes généraux :

Comme il a été décrit au chapitre précédent, chaque région résultant de la segmentation de l'image est représentée par une approximation au sens des moindres carrés au moyen d'un ensemble de fonctions polynomiales bidimensionnelles.

Si l'on part d'une image originale de taille $2^q \times 2^q$, il s'agit de mesurer la qualité de l'approximation à l'intérieur de celle-ci. Si celle-ci est insuffisante, on subdivise l'image en quatre carrés de taille $2^{q-1} \times 2^{q-1}$. On calcule ensuite pour chacun de ces carrés l'approximation au sens des moindres carrés au moyen du même ensemble de monômes $\Psi_j(x,y)$. La qualité de cette approximation est alors estimée pour chacun d'eux. Si celle-ci est acceptable, le carré correspondant apparaîtra comme définitif dans la division. Ce carré définira un espace ou une partie d'espace régulier de l'image. Si la qualité est par contre jugée insuffisante, on recommence le processus de subdivision pour le carré correspondant. Ce principe est illustré à la figure 3.1. Le première taille d'analyse ($2^q \times 2^q$) définit ce que l'on appelle le premier *niveau* de traitement. De la même façon, les tailles inférieures ($2^{q-1} \times 2^{q-1}$, $2^{q-2} \times 2^{q-2}$, ...) définissent les niveaux de traitement suivants (2, 3, ...).

La division doit nécessairement se terminer puisqu'on ne peut aller au-delà de la résolution de l'image (carré de taille 1×1 = dimension d'un pixel). En réalité, il s'arrête avant, car le polynôme bidimensionnel va interpoler les pixels originaux du carré dès que la taille de celui-ci va devenir inférieure ou égale au nombre r de coefficients du polynôme.

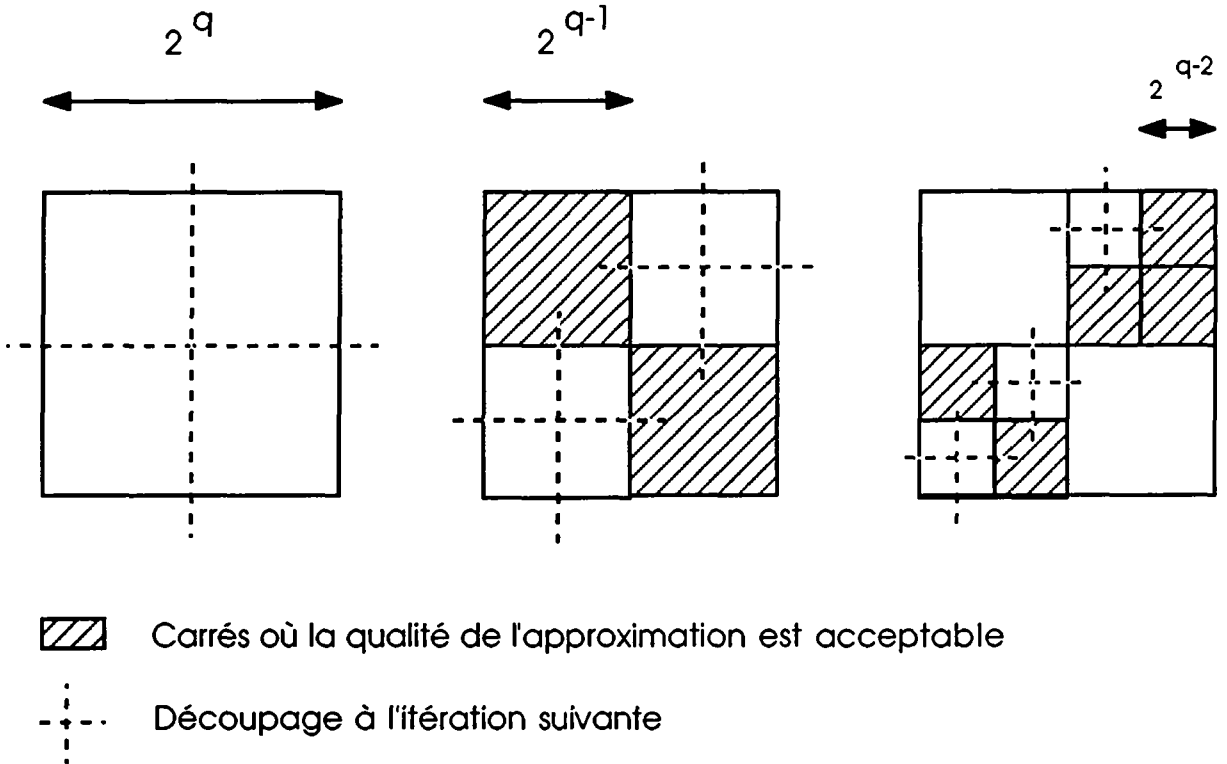


Figure 3.1: Principe de division.

Après la division, la partition de l'image est définie par une série de carrés ayant sur chaque côté un nombre de pixels constituant une puissance de 2.

La qualité de l'approximation est estimée sur la base des critères EQM et EC (voir section 2.4). Pour tenir compte de l'adéquation moyenne sur un carré donné on utilise l'erreur quadratique moyenne (EQM). Pour préserver les contours de l'image dans le carré considéré, on fait appel à l'erreur de contour (EC). Sur chacune de ces mesures seront placés des seuils d'acceptabilité, respectivement T_1 et T_2 . Dès que l'un des deux est franchi, la division sera effectuée, comme décrit précédemment. Le seuil placé à T_2 sera faible alors qu'il restera modérément élevé pour T_1 , ceci en liaison avec les hypothèses de non-dégradation de l'information contour et de dégradation tolérée pour la reproduction de l'information texture (voir paragraphe 2.1.1). La figure 3.2 reproduit l'algorithme de division dans son ensemble.

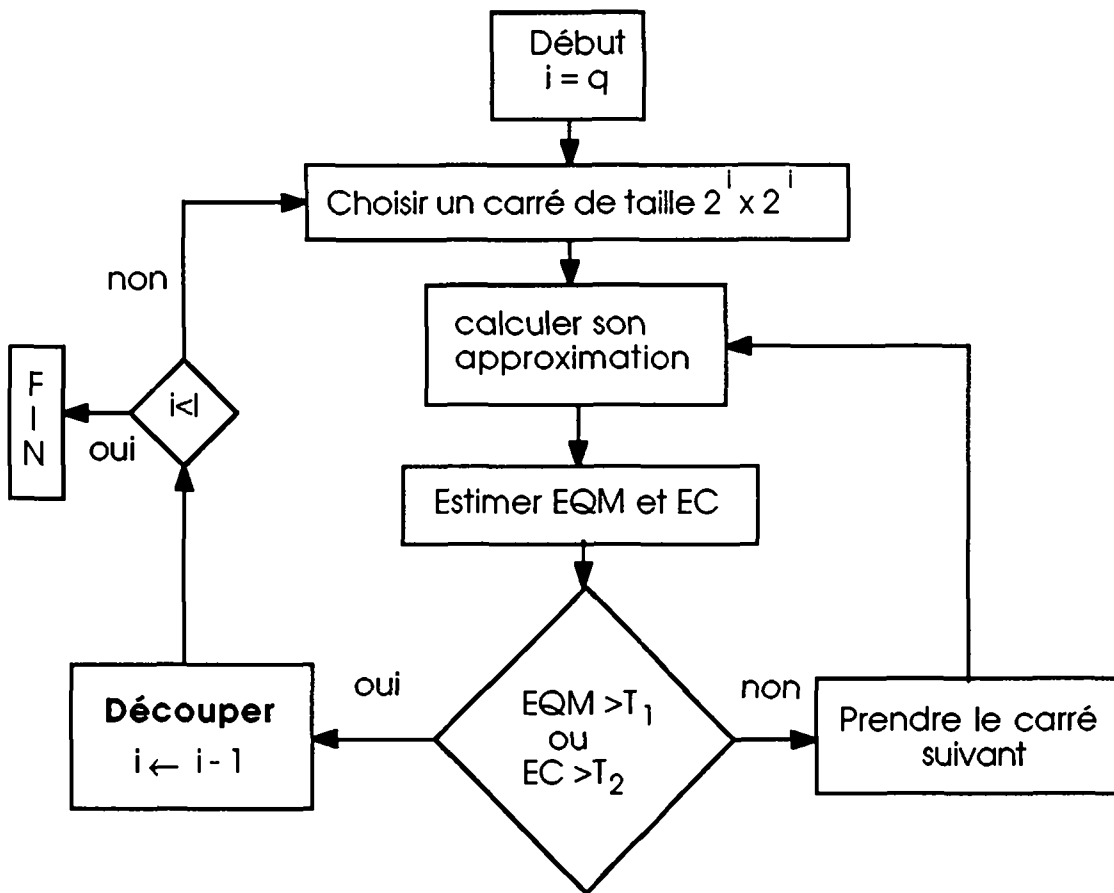


Figure 3.2: Algorithme de division (l : plus petite puissance de 2 admise selon (3.1)).

Dans les paragraphes qui suivront, différents commentaires sont apportés au principe de l'algorithme de division qui vient d'être décrit.

3.1.1 Existence d'une solution dans le calcul de l'approximation:

De manière à éviter de rendre le calcul de l'approximation impossible, la matrice \mathbf{S} devenant singulière, on a décidé de limiter le processus de subdivision à la taille de carré immédiatement supérieure à une indétermination du système (2.15). Le plus petit carré aura au plus $2^l \times 2^l$ points, avec :

$$2^l \times 2^l \leq r \quad (3.1)$$

La matrice \mathbf{S} ne présentera jamais de singularité pour des régions carrées de tailles plus grandes que $2^l \times 2^l$ si les polynômes usuels (de degré 0, 1, 2, ...) sont utilisés pour approximer les données de départ. Ceci est démontré dans l'annexe C.

3.1.2 Formes des régions après division:

Il est à priori envisageable de subdiviser l'image originale en d'autres formes géométriques que des carrés. Prendre des carrés s'explique par simplicité d'une part, par le fait que les images de départ sont en général de taille carrée et que la

grille d'échantillonnage est carrée. Disons encore que la prédominance de contours horizontaux et verticaux dans les images naturelles privilégie le choix de décomposer l'image en un ensemble de régions présentant des frontières de mêmes directions. Le principe de l'algorithme de division reste inchangé si l'on décidait de diviser l'image en des régions de même forme non carrées, par exemple des triangles isocèles rectangles (Des problèmes peuvent seulement surgir si l'on désire descendre au niveau d'un pixel pour la plus petite taille de région, celle-ci étant nécessairement carrée).

Le fait de subdiviser chaque carré en quatre sous-carrés de taille deux fois plus petite permet de passer *progressivement* d'une analyse globale de l'image à son analyse locale. Une subdivision en un nombre plus élevé de régions serait défavorable s'il existe des parties d'espaces réguliers de plus grande taille. Une division en deux rectangles identiques compliquerait l'algorithme car il faudrait tester les deux décompositions possibles de manière à choisir celle qui conduit à la meilleure reproduction des données de départ, ceci pour autant qu'un découpage ultérieur ne s'avère pas nécessaire.

Il faut que dans la partition finale, l'image soit décomposée en un ensemble de régions de n'importe quelle forme et taille, il semble nécessaire de subdiviser l'image jusqu'au niveau d'un pixel. Cette subdivision ultérieure sera réalisée uniquement pour les carrés contenant des contours en utilisant l'image de contrôle des structures positives et négatives définie au paragraphe 2.2.3. Dans le processus de rassemblement, on garantira la précision requise en traitant séparément l'information relative à ces carrés contenant des contours.

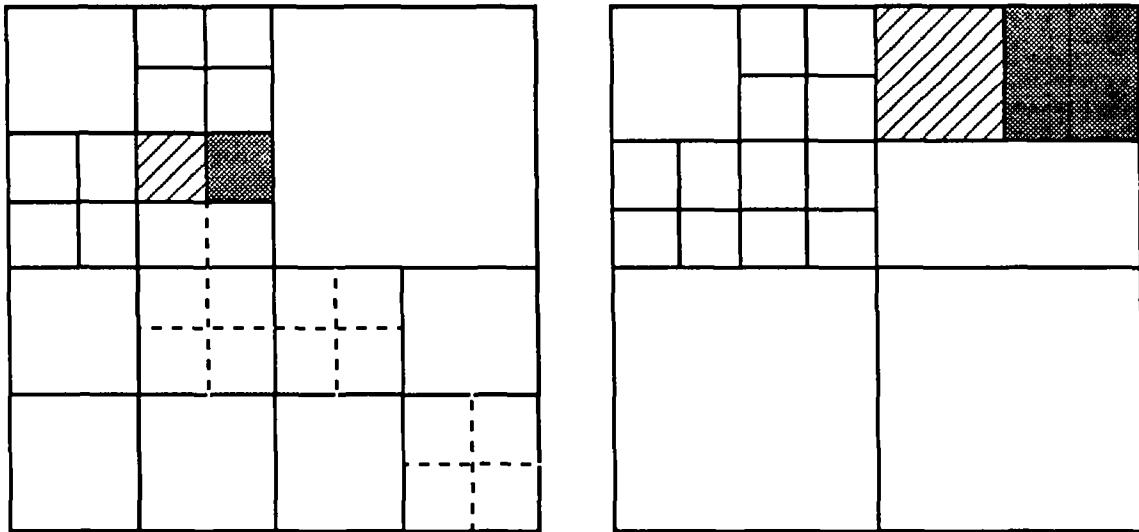
Après avoir décrit les principes de la division, la section suivante présente la structure des données nécessaire à une implantation de l'algorithme.

3.2 Implantation de l'algorithme:

Deux solutions sont proposées. La première solution permet de traiter tous les carrés d'un même niveau simultanément. La seconde fait appel à la notion de récursivité. L'analyse de chaque carré est alors effectuée cycliquement du premier niveau (image) au niveau le plus élevé (plus petite taille de carré) selon un balayage prédéfini.

Dans le premier cas, on traite tous les carrés d'une même taille simultanément alors que dans le deuxième cas, on cherche systématiquement à diviser un carré avant de considérer un autre carré.

La figure 3.3a illustre un état intermédiaire de l'algorithme lorsqu'on utilise la première possibilité; La figure 3.3b fait de même si l'on choisit la méthode récursive.



a) Analyse hiérarchique (les carrés à analyser de même taille que le carré courant sont indiqués en traitillé).

b) Analyse descendante-ascendante cyclique.

Figure 3.3: Etat intermédiaire de l'algorithme de division.
Le carré couramment analysé est indiqué en hachuré.
Celui qui sera analysé à l'étape suivante est indiqué en gris.

Ces deux approches donnent les mêmes résultats; elles ne définissent que deux implantations différentes de l'algorithme.

3.2.1 Analyse hiérarchique:

On commence l'analyse au premier niveau, c'est-à-dire $2^q \times 2^q$. Si une division est nécessaire, on mémorise dans une liste les coordonnées respectives du premier point en haut à gauche de chaque carré résultant de la subdivision, en l'occurrence $(1,1)$, $(1,1+2^{q-1})$, $(1+2^{q-1},1)$ et $(1+2^{q-1},1+2^{q-1})$. Ces coordonnées vont caractériser le carré. On répète alors la procédure pour tous les carrés restant à analyser. Si une division s'avère nécessaire, on insère *en fin de liste*, les coordonnées associées aux quatre carrés de taille immédiatement inférieure résultant de la subdivision du carré analysé. A chaque fois qu'un carré est analysé, on l'élimine de la liste. L'algorithme se termine dès que la liste est vide. De manière à savoir quelle est la dimension d'un carré dans la liste, il suffit d'utiliser deux compteurs. Le premier indique le nombre de carrés de taille donnée restant à analyser dans la liste, le second, le nombre de carrés de taille moitié en fin de liste. Chaque fois qu'un carré est analysé le premier compteur est décrémenté d'une unité; s'il y a division, le second est incrémenté de quatre unités. Dès que le premier compteur arrive à zéro, on échange les valeurs des deux compteurs. On sait qu'à partir de ce moment on travaille avec des carrés de taille moitié de la taille qu'on a eu jusqu'à ce que le premier compteur arrive à zéro. Dans l'exemple de la figure 3.3a, après l'analyse du carré hachuré, le premier compteur prend la valeur 15 (nombre de carrés en traitillé+carré en gris). En utilisant un tel algorithme, on remarque que l'analyse se fait par taille de carré d'où le nom d'analyse hiérarchique.

3.2.2 Analyse descendante-ascendante cyclique:

L'analyse est effectuée ici du niveau global au niveau local de manière cyclique. Une approche récursive est présentée par l'organigramme de la figure 3.4.

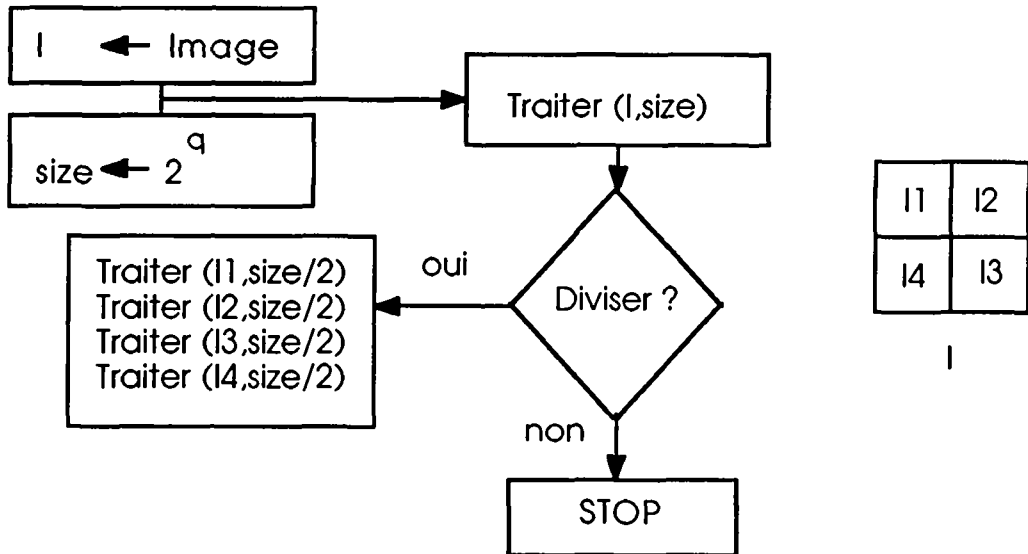


Figure 3.4: Implantation récursive de l'algorithme de division avec analyse descendante-ascendante cyclique.

Sans faire appel à la notion de récursivité, on peut définir une implantation par analyse cyclique en utilisant la structure de liste du paragraphe 3.2.1. Il suffit pour cela d'insérer dans la liste les coordonnées des carrés de taille immédiatement inférieure *juste après la position courante* du carré analysé. Dans ce cas, on n'a plus besoin de mémoriser de compteurs puisque l'on sait qu'un découpage entraînera seulement une insertion de quatre carrés de taille immédiatement inférieure. On peut également appliquer cet algorithme tel quel à des images de dimension quelconque. Il suffit simplement de modifier la liste en conséquence avant de commencer l'analyse.

Nous verrons à la section suivante que les opérations réalisées à un même niveau peuvent être grandement simplifiées par un choix adéquat de l'origine. Cette considération a justifié le choix d'une analyse hiérarchique dans la réalisation pratique de l'algorithme de division. Une fois le mode d'implantation choisi, il s'agit de décrire dans ce qui suit la complexité de chaque phase de l'algorithme.

3.3 Complexité de l'algorithme de division :

La complexité est liée principalement aux calculs de l'approximation optimale au sens des moindres carrés. Nous verrons comment la forme carrée des régions après division ainsi que le choix de fonctions de base polynomiales permettent de simplifier considérablement le problème. Il sera également montré comment cette

phase de la segmentation peut être traitée à partir de fonctions polynomiales séparables.

Des considérations sur les images naturelles traitées permettront de limiter le nombre de carrés à étudier pour aboutir à la partition après division et de simplifier les critères d'erreur choisis.

3.3.1 Aspects liés à l'approximation:

Le premier point à traiter consiste à choisir l'origine de l'espace bidimensionnel par rapport à laquelle seront effectués tous les calculs. Il est apparu avantageux de fixer une origine par rapport à chaque carré tout au long de l'opération de division. En effet, si l'on part de l'expression (2.15) qui représente le système à résoudre pour le calcul optimal des coefficients, on remarque que l'expression des matrices **Z** et par conséquent **S** n'est liée qu'aux positions des points du domaine **D** et non pas à la valeur du signal en ces points. Deux domaines, dont l'un peut être obtenu par translation géométrique de l'autre, auront donc les mêmes matrices **Z** et **S** associées si l'on choisit une origine identique relativement à chaque domaine (par exemple, son centre de symétrie). Ainsi, on a réduit le calcul de l'approximation pour chaque région à l'estimation du vecteur **h** et à la résolution du système **Su = h**, ce qui revient à $O(\max(Nr, r^3))$ opérations, où **N** représente le nombre de points du domaine et **r** le nombre de fonctions de base. Ce principe s'applique parfaitement à l'algorithme de division vu qu'à chaque niveau, on traite des carrés de même taille.

Pour chaque niveau de division, on n'estimera qu'une fois l'expression des matrices **S** et **Z**. Celles-ci seront donc évaluées au plus q fois tout au long de l'algorithme (l'image a une taille initiale $2^q \times 2^q$).

Le deuxième point est lié à l'expression de la matrice **S** si l'on utilise des fonctions polynomiales pour approximer les données de départ. L'annexe C montre que chaque terme de cette matrice peut être exprimé par un produit de deux séries de coordonnées élevées à une certaine puissance (voir (c1)). Appelons (k_0, l_0) les coordonnées de l'origine relative à un carré de taille donnée. D'une façon générale, chaque série associée à la coordonnée **k** sera représentée par:

$$\sum_{k=1}^K (k-k_0)^m = \sum_{k=1}^K \left(\sum_{n=0}^m \binom{m}{n} k^n \cdot k_0^{m-n} \right) \quad (3.2)$$

où **m** représente un nombre entier positif et **K** correspond au nombre de points sur le côté du carré, c'est-à-dire une puissance de 2. En intervertissant les deux opérateurs de somme, on obtient :

$$\sum_{k=1}^K (k-k_0)^m = \sum_{n=0}^m \binom{m}{n} k_0^{m-n} \cdot \sum_{k=1}^K k^n \quad (3.3)$$

Une expression semblable est obtenue en remplaçant k et k_0 par l et l_0 respectivement pour la série relative à la coordonnée l . Dans la formule (3.3), on voit apparaître la somme de K entiers consécutifs élevés à la puissance m . La valeur de m la plus élevée apparaissant dans la matrice \mathbf{S} correspond à p^2 si p est le degré du polynôme considéré. En se limitant aux polynômes bidimensionnels de degré 3, on ne considèrera que les séries entières de puissances 0 à 6, notées par σ_0 à σ_6 . Leurs expressions sont données par [54, p.36]:

$$\begin{aligned}\sigma_0 &= K & \sigma_1 &= K(K+1)/2 & \sigma_2 &= \sigma_1(2K+1)/3 \\ \sigma_3 &= \sigma_1^2 & \sigma_4 &= \sigma_2(3K^2+3K-1)/5 & & (3.4) \\ \sigma_5 &= \sigma_3(2K^2+2K-1)/3 & \sigma_6 &= \sigma_2(3K^4+6K^3-3K+1)/7\end{aligned}$$

A partir de ces relations, il est très simple d'évaluer l'expression (3.3). On peut simplifier ultérieurement le calcul de la matrice \mathbf{S} en choisissant pour origine le centre de symétrie du carré analysé. En effet, dans ce cas, toutes les séries correspondant à des valeurs impaires de m sont nulles (voir annexe C). La matrice \mathbf{S} prend alors l'expression (c2) pour des polynômes de degré 3 et se limite aux sous-matrices de taille 1x1, 3x3 et 6x6 inscrites dans la partie supérieure gauche de la matrice représentée par (c2) pour des polynômes de degré 0, 1 et 2, respectivement.

Les termes a , b , c et d de l'annexe C s'estiment pour un carré de taille donnée à partir des relations (3.3).et (3.4). En utilisant un programme de manipulations d'expressions symboliques, il est encore possible de faciliter la tâche de la résolution du système (2.15) en explicitant l'expression des inverses des matrices \mathbf{S} . Appelons \mathbf{S}_0 , \mathbf{S}_1 , \mathbf{S}_2 et \mathbf{S}_3 , les matrices correspondant à chaque degré de polynôme. On peut écrire:

$$\mathbf{S}_0^{-1} = (1/K^2) \quad (3.5)$$

$$\mathbf{S}_1^{-1} = \begin{bmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/(ab) & 0 \\ 0 & 0 & 1/(ab) \end{bmatrix} \quad (3.6)$$

$$\mathbf{S}_2^{-1} = \begin{bmatrix} \alpha_2 & 0 & 0 & \beta_2 & 0 & \beta_2 \\ 0 & \gamma_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_2 & 0 & 0 & 0 \\ \beta_2 & 0 & 0 & \delta_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/b^2 & 0 \\ \beta_2 & 0 & 0 & 0 & 0 & \delta_2 \end{bmatrix} \quad (3.7)$$

où $\delta_2 = 1/(ac-b^2)$, $\alpha_2 = (ac+b^2)/(a^2\delta_2)$, $\beta_2 = -b/(a\delta_2)$ et $\gamma_2 = 1/(ab)$.

$$\mathbf{S}_3^{-1} = \begin{bmatrix} \alpha_3 & 0 & 0 & \beta_3 & 0 & \beta_3 & 0 & 0 & 0 & 0 \\ 0 & \gamma_3 & 0 & 0 & 0 & 0 & \delta_3 & 0 & \varepsilon_3 & 0 \\ 0 & 0 & \gamma_3 & 0 & 0 & 0 & 0 & \varepsilon_3 & 0 & \delta_3 \\ \beta_3 & 0 & 0 & -\varepsilon_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/b^2 & 0 & 0 & 0 & 0 & 0 \\ \beta_3 & 0 & 0 & 0 & 0 & -\varepsilon_3 & 0 & 0 & 0 & 0 \\ 0 & \delta_3 & 0 & 0 & 0 & 0 & \eta_3 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon_3 & 0 & 0 & 0 & 0 & \kappa_3 & 0 & 0 \\ 0 & \varepsilon_3 & 0 & 0 & 0 & 0 & 0 & 0 & \kappa_3 & 0 \\ 0 & 0 & \delta_3 & 0 & 0 & 0 & 0 & 0 & 0 & \eta_3 \end{bmatrix} \quad (3.8)$$

où $\alpha_3 = \alpha_2$, $\beta_3 = \beta_2$, $\delta_3 = c/(a(c^2 - bd))$, $\varepsilon_3 = -\delta_2$, $\gamma_3 = \delta_2 \delta_3 (ad - bc)$, $\eta_3 = b/(a(bd - c^2))$ et $\kappa_3 = a\delta_2/b$.

Comme l'expression des matrices ci-dessus reste inchangée pour toute image traitée, il suffit de les mémoriser pour toutes les tailles possibles de carrés. Pour justifier le gain en calcul qui peut ainsi être obtenu, le tableau 3.5 compare le nombre d'opérations nécessaires pour des polynômes de différents degrés sur une certaine image. Prenons par exemple, l'image du caméraman. L'estimation reste valable pour d'autres catégories d'images car on peut observer expérimentalement que le nombre de carrés de chaque taille est du même ordre de grandeur d'une image à l'autre. Après application de l'algorithme de division en utilisant un polynôme de degré 3, l'image du caméraman est formée d'un carré de taille 64x64, de 11, 33, 130 et 2088 carrés de taille 32x32, 16x16, 8x8 et 4x4 respectivement. Pour arriver à ce résultat, il a fallu analyser 16, 60, 196, 652 et 2088 carrés de tailles respectives 64x64, 32x32, 16x16, 8x8 et 4x4, en supposant qu'on ait commencé l'algorithme de division à partir de 16 carrés 64x64. Une estimation de la complexité de calcul devient dès lors possible. Un raisonnement similaire peut être utilisé pour des polynômes de degré inférieur. Le cas des polynômes de degré 0 ne sera pas pris en considération ici, car des simplifications ultérieures de l'algorithme peuvent être utilisées. Dans l'estimation de la complexité pour une origine placée au centre de symétrie de chaque carré (origine locale), on a utilisé les relations (3.6) à (3.8) pour le calcul du produit $\mathbf{S}^{-1}\mathbf{h}$, ce qui peut être obtenu en environ 3, 10 et 22 opérations pour des polynômes de degré 1, 2 et 3 respectivement. Si l'on ne considère qu'une origine unique (origine globale), le calcul de la complexité est alors réduit aux étapes 2 à 4 décrites au paragraphe 2.3.1.

m \	Origine locale	Origine globale
1	840	2 770
2	1530	7510
3	2 560	19 450

Tableau 3.5: Estimation de la complexité de calcul de l'algorithme de division.
Les valeurs indiquées sont en milliers d'opérations nécessaires.
(m=degré du polynôme).

Rôle de la séparabilité:

Il est possible de limiter la complexité de l'algorithme de division par le choix d'un ensemble de fonctions polynomiales séparables. Il est en réalité plus correct de dire que la solution du problème de l'approximation au sens des moindres carrés est rendue séparable pour des régions de forme rectangulaire par ce choix adéquat de fonctions de base. D'une façon générale, de tels polynômes sont décrits par la relation:

$$g^{\wedge}(x,y) = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} u_{ij} x^{i-1} y^{j-1} \tag{3.9}$$

Si l'on désire approximer une région de forme rectangulaire constituée par un ensemble de K lignes et L colonnes, la région approximée qui peut être notée par une matrice G^{\wedge} de taille KxL, pourra être définie au moyen de la relation:

$$G^{\wedge} = V_k.U.V_l^T \tag{3.10}$$

où V_k et V_l sont des pseudo-matrices de Vandermonde de taille respective $K \times r_1$ et $L \times r_2$. Chaque ligne de coordonnée x d'une pseudo-matrice de Vandermonde V de taille $N \times r$ s'exprime par le vecteur $[1 \ x \ x^2 \ \dots \ x^{r-1}]$. La somme des erreurs quadratiques d'approximation peut-être exprimée par:

$$\begin{aligned} EQ &= \sum_{k=1}^K \sum_{l=1}^L (g(k,l) - g^{\wedge}(k,l))^2 \\ &= \text{Trace} ((G - G^{\wedge})^T.(G - G^{\wedge})) \end{aligned} \tag{3.11}$$

où \mathbf{G} représente le tableau des données originales. La solution qui minimise cette erreur est obtenue par annulation de la dérivée $\partial EQ / \partial \mathbf{U}$. Elle satisfait l'équation [55]:

$$\mathbf{U} = (\mathbf{V}_k^T \mathbf{V}_k)^{-1} \cdot \mathbf{V}_k^T \cdot \mathbf{G} \cdot \mathbf{V}_l \cdot (\mathbf{V}_l^T \mathbf{V}_l)^{-1} \quad (3.12)$$

Pour simplifier la notation, nous appellerons \mathbf{S}_k la matrice carrée de taille $r_1 \times r_1$ $\mathbf{V}_k^T \mathbf{V}_k$; \mathbf{S}_l la matrice carrée de taille $r_1 \times r_1$ $\mathbf{V}_l^T \mathbf{V}_l$; enfin, la matrice $\mathbf{V}_k^T \cdot \mathbf{G} \cdot \mathbf{V}_l$ de taille $r_1 \times r_2$ sera appelée \mathbf{H} . On remarque que les matrices \mathbf{S}_k et \mathbf{S}_l sont de type Hankel. En effet, elles ont la forme:

$$\begin{bmatrix} v_0 & v_1 & \dots & v_{r-1} \\ v_1 & v_2 & \dots & v_r \\ | & | & \dots & | \\ v_{r-1} & v_r & \dots & v_{2r} \end{bmatrix} \quad (3.13)$$

Cette matrice est entièrement spécifiée par $2r+1$ termes, qui dans le cas des fonctions polynomiales choisies s'expriment pour \mathbf{S}_k par:

$$v_i = \sum_{k=1}^K k^i \quad (3.14)$$

La résolution du système (3.12) est beaucoup plus facile à obtenir que dans le cas d'un polynôme de forme générale. En effet, celle-ci nécessite:

1.- l'inversion de deux matrices de Hankel de tailles respectives $r_1 \times r_1$ et $r_2 \times r_2$, ce qui peut être effectué en $O(r_1^2)$ et $O(r_2^2)$ opérations au moyen d'algorithmes appropriés [56].

2.- cinq multiplications matricielles. Si l'on considère des carrés de taille txt , ces multiplications peuvent être obtenues en $\min(r_1, r_2) \times \text{txt}^2 + r_1 \times r_2 \times \text{txt} + r_1^2 \times r_2 + r_1 \times r_2^2$ opérations, pour autant que celles-ci soient effectuées dans un ordre correct. Dès que le produit $\min(r_1, r_2)$ est plus grand que 1, cette approche est favorable par rapport à la formulation non séparable quelle que soit la valeur de t . Ceci est montré dans l'annexe D.

Pour ce qui est de l'expression de l'erreur quadratique, celle-ci peut également être obtenue sans effectuer le calcul complet des valeurs approximées représentées par la matrice \mathbf{G}^\wedge . Elle s'exprime par (voir annexe E):

$$EQ = \text{Trace}(\mathbf{G}^T \mathbf{G}) - \text{Trace}(\mathbf{U}^T \mathbf{H}) \quad (3.15)$$

Elle peut s'obtenir pour un carré de taille txt en environ $t^2 + \min(r_1 + r_2^2, r_1^2 + r_2)$ opérations (voir annexe E). Cette approche séparable s'avère de nouveau plus efficace dès que r_1 et r_2 sont plus grands que 1.

Dans le contexte de l'algorithme de division, les considérations relatives au choix de l'origine sont indépendantes du choix des fonctions de base et s'appliquent donc parfaitement aux polynômes séparables. En plaçant celle-ci au centre de symétrie de chaque carré, on rend bon nombre d'éléments des matrices de Hankel S_k et S_l nuls (tous les termes correspondant à des valeurs de i impaires). Ceci permet de simplifier une nouvelle fois la complexité de l'algorithme de division en limitant les calculs liés aux approximations moindres carrés.

Le tableau 3.6 compare pour différentes valeurs de $r_1 \times r_2$ et t la complexité de calcul liée à une formulation séparable et non séparable du problème d'approximation. On a utilisé les expressions de N_s et N_{ns} données dans l'annexe E, où l'on a négligé les opérations relatives aux inversions de matrice. En effet, celles-ci peuvent être considérablement réduites par un choix adéquat de l'origine comme centre de symétrie de chaque carré.

<div>taille poly</div>	$t = 4$	$t = 8$	$t = 16$	$t = 32$	$t = 64$
$r_1 = 2$ $r_2 = 2$	64 80	176 272	592 1040	2192 4112	8464 16400
$r_1 = 2$ $r_2 = 3$	86 132	206 420	638 1572	2270 6180	8606 24612
$r_1 = 2$ $r_2 = 4$	112 192	240 576	688 2112	2352 8256	8752 32832
$r_1 = 3$ $r_2 = 3$	138 225	318 657	966 2385	3414 9297	12918 36945
$r_1 = 3$ $r_2 = 4$	180 336	372 912	1044 3216	3540 12432	13140 49296

Tableau 3.6: Comparaison du nombre d'opérations nécessaires au calcul de l'approximation polynomiale séparable entre une formulation séparable et non séparable du problème d'approximation. Pour chaque valeur des paramètres, la ligne supérieure correspond à N_s , la ligne inférieure à N_{ns} .

3.3.2 Simplifications rendues possibles grâce à la structure des images traitées:

Les images traitées sont supposées contenir de l'information même détaillée. Il est donc raisonnable de penser qu'une scène d'intérêt apparaît fréquemment au centre de l'image, vu que l'on tâche de cadrer le plus souvent la scène. L'application de l'algorithme de division ne peut en général pas donner de carré de taille supérieure à 64x64 pour une image de départ de taille 256x256. Ceci justifie de commencer l'application de l'algorithme de division à un niveau relativement élevé, d'autant plus qu'on utilise un nombre de coefficients très restreint pour

reconstituer le signal sur un grand domaine. Aussi, on a choisi de faire débiter l'analyse avec des carrés de taille 64x64.

Il apparaît vraisemblable que pour chaque carré de grande taille analysé au cours du processus de division, l'erreur de contour (EC) soit systématiquement plus grande que le seuil T_2 que l'on a fixé pour éviter de perdre de l'information, ceci à cause de l'impossibilité de représenter un saut de luminance au moyen d'une approximation polynomiale bidimensionnelle calculée selon le critère des moindres carrés lorsque le nombre de points du domaine analysé est beaucoup plus grand que le nombre de coefficients du polynôme. Une subdivision du carré en question est dès lors jugée nécessaire. On peut donc choisir de mesurer simplement la présence d'un contour, pour décider s'il faut subdiviser ultérieurement le carré. Le calcul de l'approximation optimale puis de l'erreur quadratique moyenne (EQM) n'est alors effectué que si le carré analysé ne présente pas de contours. Ceci permet de juger de la correspondance moyenne entre les données de départ et celles approximées. Pour les petites tailles de carré, c'est-à-dire lorsque le nombre de pixels est sensiblement égal au nombre de coefficients du polynôme, on peut vérifier qu'un contour contenu à l'intérieur du carré en question peut être correctement reproduit. Dans ce cas, il est également inutile de mesurer l'erreur de contour (EC) car elle s'apparente relativement bien à l'erreur quadratique moyenne (la plupart des points du carré correspondant à des points situés de part et d'autre du contour). Grâce à ces considérations, il est ainsi possible d'éviter bon nombre de calculs d'approximation et d'éliminer la mesure d'erreur de contour EC. L'algorithme de division de la figure 3.2 peut donc se simplifier comme indiqué à la figure 3.7 sans distorsion complémentaire.

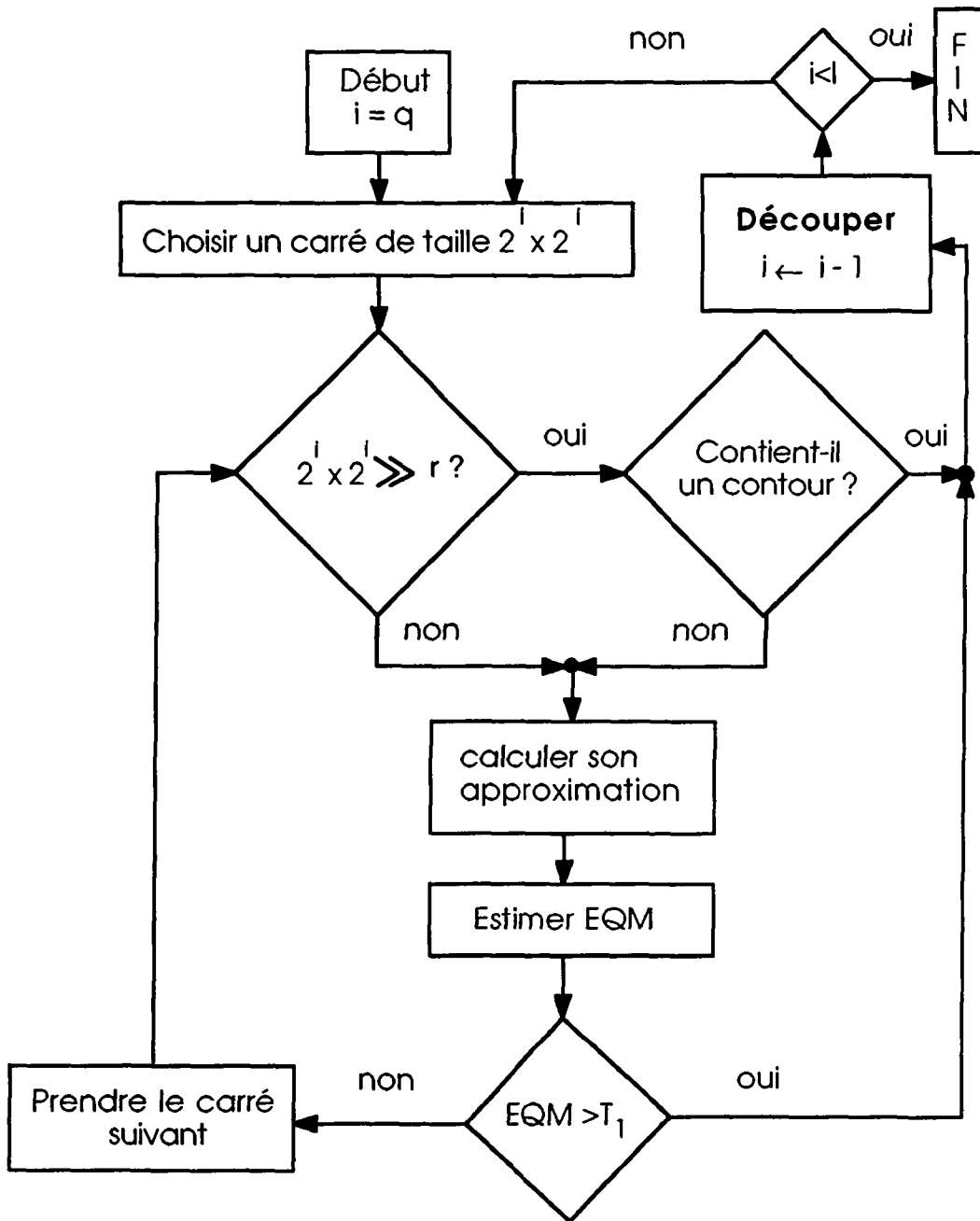


Figure 3.7: Algorithme de division simplifié par analyse de présence des contours (l: dernier niveau de division défini par la relation (3.1)).

Une fois l'image analysée au moyen de l'algorithme de division, nous allons étudier dans la section qui suit comment représenter la segmentation en carrés ainsi obtenue.

3.4 Représentation de l'information segmentée et postraitement :

Si l'on considère la segmentation obtenue après application de l'algorithme de division, on peut reconstruire une image approximée en connaissant la structure de l'image divisée et les coefficients d'approximation polynomiaux pour chaque carré. Le premier paragraphe illustre comment représenter l'information de position, le second indique comment il est possible de quantifier l'information relative aux coefficients polynomiaux. Ceci permettra d'estimer la compression que l'on peut atteindre à ce niveau du traitement de l'algorithme de division et rassemblement. Enfin, le dernier paragraphe décrit le genre de postraitements qui peuvent être utilisés de manière à rehausser la qualité de l'image reconstruite après application de l'algorithme de division.

3.4.1 Position et taille des carrés:

Dans ce paragraphe, deux méthodes sont proposées pour coder la position et la taille des carrés obtenus par l'algorithme de division. La première est basée sur la structure d'arbre quaternaire qui peut être utilisée pour représenter la partition de l'image en carrés. La seconde est un peu plus complexe mais permet de générer la partition avec un nombre plus réduit de bits.

Codage par arbre quaternaire:

Le graphe de la segmentation représentant l'image divisée peut se ramener à un arbre quaternaire. La figure 3.8b illustre la représentation en arbre quaternaire de la partition de la figure 3.8a.

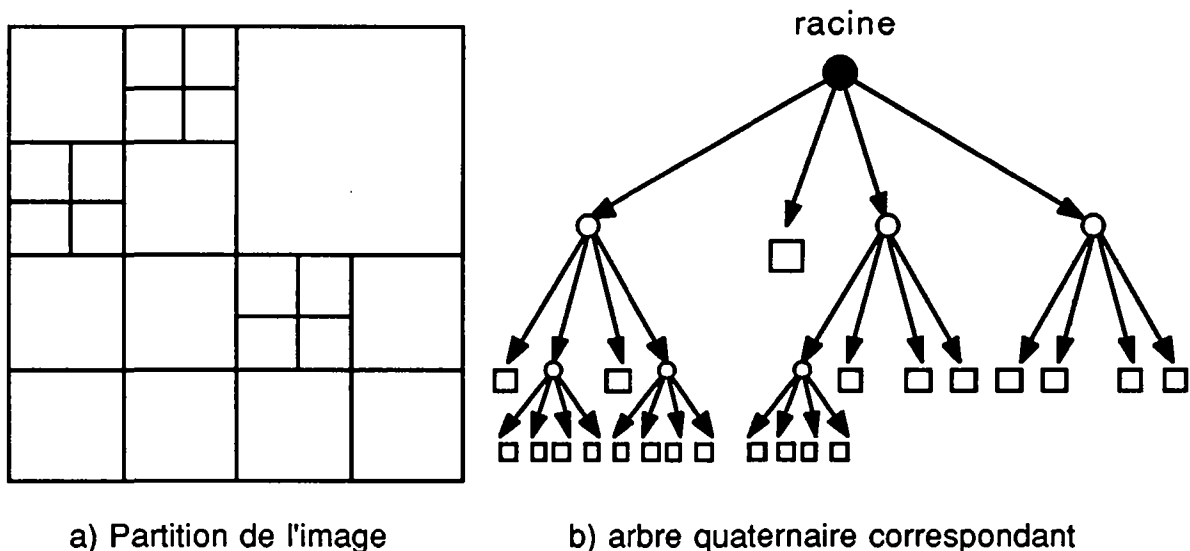


Figure 3.8: Représentation par arbre quaternaire.

Les arbres quaternaires définissent une structure de données hiérarchique compacte. Leur utilisation n'est pas forcément liée à la segmentation d'images [57,25]. Dans le cadre des applications pour la compression d'images, ils ont l'avantage de permettre une transmission progressive de l'information. Récemment, Cohen et al. [58] ont analysé, au moyen d'arbres quaternaires, la compression que l'on pouvait atteindre pour le codage hiérarchique d'images binaires.

Le principe de codage qui peut être obtenu pour représenter la structure des carrés obtenus après division est relativement simple. En partant de la racine de l'arbre, on transmet un bit pour dire si l'on peut descendre à un niveau plus bas, ou si l'on se trouve sur une feuille de l'arbre. Une autre manière de voir les choses consiste à transmettre un bit pour déterminer s'il est nécessaire de diviser le carré analysé ou pas. On peut éviter de transmettre le bit correspondant au dernier niveau de l'arbre si l'on sait qu'une subdivision ultérieure est impossible (taille du pixel atteinte ou nombre de pixels du carré immédiatement supérieur au nombre de coefficients d'approximation). Bien entendu, ceci présuppose que le récepteur connaît la taille de l'image à transmettre et le nombre de coefficients d'approximation. De la même façon, on peut commencer le code à un certain niveau dans l'arbre, si l'on sait que l'image ne peut présenter de carrés de taille plus grande que 64×64 (cf. paragraphe 3.3.2). De nouveau, on suppose que cette information est connue du récepteur.

Le coût le plus élevé que l'on puisse obtenir avec un tel schéma de codage correspond à une partition constituée uniquement de carrés de taille la plus petite possible, soit $2^l \times 2^l$. Le coût le moins élevé consiste à avoir des carrés de taille 64×64 uniquement. En appelant C un tel coût exprimé en bits, on a la relation (en supposant $q \geq 6$):

$$2^{2(q-6)} \leq C \leq \sum_{i=l+1}^q 2^{2i} = (4^{q+1} - 4^{l+1})/3 \quad (3.16)$$

La section suivante illustrera pratiquement le coût nécessaire pour les trois images traitées selon divers degrés du polynôme. On remarquera que le coût est souvent proche de la borne supérieure car le nombre de carrés de taille $2^l \times 2^l$ est très élevé.

Ce code même s'il permet une transmission hiérarchique, par taille de carré, ne s'adapte pas bien à des images rectangulaires de dimension quelconque. En outre, il utilise un faible nombre de bits pour transmettre un carré de grande taille. Or de tels carrés apparaissent rarement après application de l'algorithme de division. Ceci va à l'encontre de la philosophie de coût minimum qui tend à associer un mot long à un symbole rare et inversement. Néanmoins, ce code tient efficacement compte des contraintes géométriques imposées par l'algorithme de division et par la taille de l'image traitée. Pour tenir compte de toutes ces observations, on a défini un code un peu plus efficace sans que sa complexité en soit sensiblement accrue. Chaque fois qu'un carré est transmis, les contraintes géométriques de l'algorithme de division permettent de modifier la table de correspondance de chaque symbole utilisé.

Codage conditionnel par taille de carré:

Dans cette approche, on choisit pour symbole la taille de chaque carré. En effet, en transmettant successivement la taille de chaque carré selon un ordre de balayage prédéfini, il est possible de retrouver la partition donnée par l'algorithme de division. Par la suite, on supposera que le sens de parcours ira de gauche à droite et de haut en bas.

De manière à trouver un code efficace pour chacun de ces symboles, on estime leur probabilité d'apparition p_6 à p_1 en supposant que les carrés ont des tailles allant de $2^1 \times 2^1$ à 64×64 ainsi qu'on en a fait l'hypothèse au paragraphe précédent. Il existe donc 7-1 symboles à coder que l'on notera 1 à 6. Pour la plupart des images réelles testées, le nombre de carrés de taille $2^i \times 2^i$ est beaucoup plus faible que le nombre de carrés $2^{i-1} \times 2^{i-1}$. On peut dès lors admettre que la probabilité d'apparition de n'importe quel symbole correspondant à des carrés de taille $2^i \times 2^i$ à 64×64 ($\sum_{n=i..6} p_n$) reste plus faible que la probabilité d'apparition du symbole $2^{i-1} \times 2^{i-1}$. Le code optimal de Huffman utilisant un alphabet binaire associera 1 bit au au symbole 1, 2 bits au symbole 1+1, ... , 5-1 bits au symbole 4, 6-1 bits aux symboles 5 et 6.

Pour améliorer les performances en compression, il suffit de tenir compte du fait que l'on ne peut pas avoir n'importe quelle séquence de symboles 1 à 6 à cause des contraintes géométriques imposées par la structure de l'image et l'algorithme de division. En effet, supposons d'avoir transmis le symbole 5 correspondant à un carré de taille 32×32 . S'il est inscrit dans la partie supérieure gauche d'un carré de plus grande taille (64×64) analysé au cours de l'application de l'algorithme, on peut en déduire que tous les carrés qui lui sont adjacents à sa droite et au-dessous de lui ainsi que selon la diagonale descendante selon une mesure de contiguïté 8-connexes ne peuvent avoir qu'une taille inférieure ou égale à lui-même, en l'occurrence 32×32 . D'une façon plus générale, considérons la figure 3.9.

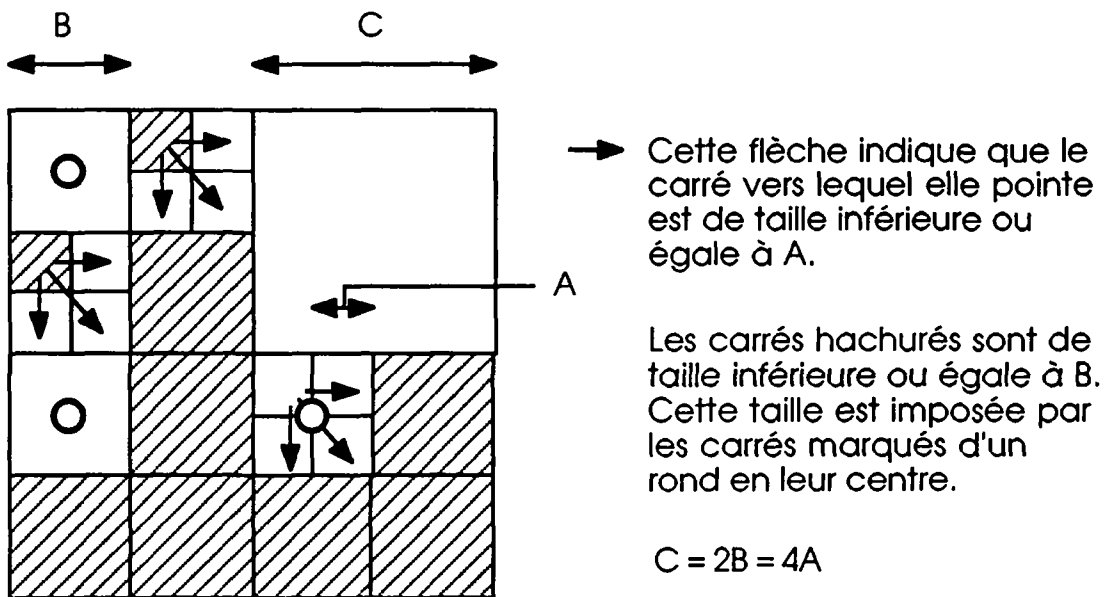


Figure 3.9: Contrainte géométrique imposée par l'algorithme de division.

Comme la contrainte géométrique imposée par l'algorithme de division agit selon une mesure de voisinage 8-connexes, on considèrera cette mesure de connexité dans ce paragraphe. Supposons vouloir transmettre la taille d'un certain carré. Regardons qu'elle est la taille que nous notons t du plus petit carré situé à sa gauche et au dessus de lui ainsi que selon la diagonale ascendante selon la mesure de contiguïté 8-connexes. Par exemple, dans la figure 3.9, le premier carré hachuré a pour voisin de gauche un carré de taille t égale à B. La structure de l'algorithme de division impose pour ce carré hachuré de ne pas avoir de taille supérieure à B. Ceci n'est valable que parce que le carré hachuré est inscrit dans

un carré de taille C égale à $2B$, analysé lors de l'application de l'algorithme de division. D'une façon générale, le carré courant est de taille inférieure ou égale à t s'il est inscrit dans un carré de taille $2t$ analysé lors de l'algorithme de division. En particulier, dans le cas du symbole l (carré de taille $2^l \times 2^l$), cela veut dire que les 3 symboles représentant ses voisins à sa droite, au-dessous de lui ainsi que selon la diagonale descendante selon une mesure de connexité d'ordre 8 seront égaux à l .

La contrainte géométrique peut être exprimée différemment en reprenant la représentation de l'algorithme de division par arbre quaternaire. La transmission de la taille du premier carré en haut à gauche de l'image impose quel est le niveau de la branche la plus à gauche de l'arbre. Dès lors, il existe 3 noeuds de même niveau à la droite de cette branche. Il suffit de transmettre si ces noeuds correspondent à des feuilles de l'arbre ou s'ils donnent naissance à de futures branches aboutissant à des noeuds de niveau inférieur. On parcourt ainsi l'arbre de gauche à droite en transmettant le niveau de la prochaine branche.

On peut donc admettre de réduire la table de correspondance entre symboles à coder et mots binaires, selon les symboles déjà transmis. Si l'on veut connaître la taille d'un certain carré, sachant qu'il est forcément de taille inférieure ou égale à $2^i \times 2^i$ du fait des contraintes géométriques imposées par l'algorithme de division, le symbole à transmettre utilisera une table de correspondance identique à la précédente jusqu'à $i-1$ et utilisant le même nombre de bits que pour $i-1$ pour le symbole i . A la limite si i est égal à 1, aucun bit ne sera transmis et si i est égal à 6, la table restera inchangée. Ceci est illustré à la figure 3.10. Dans cette figure, les carrés blancs, gris foncé et gris clair utilisent une table de correspondance dont le mot binaire le plus long est de 4, 3 et 2 bits, respectivement.

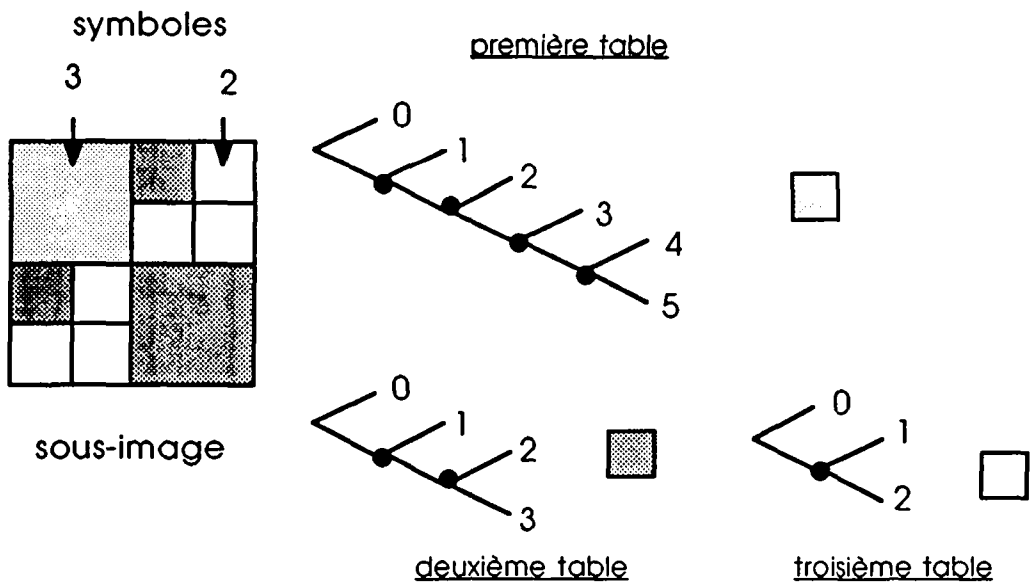


Figure 3.10: Changement de table de correspondance.
(le remplissage des carrés indique la table à utiliser)

Selon la partition obtenue, il peut être avantageux de transmettre les différents symboles selon une autre direction de balayage tout en gardant la même philosophie de codage (gauche à droite+haut en bas; gauche à droite+bas en haut; droite à gauche+haut en bas; droite à gauche+bas en haut; haut en bas+gauche à droite...). Il existe huit possibilités correspondant chacune à un certain coût de

transmission. Il suffit de transmettre trois bits supplémentaires au récepteur pour indiquer la direction correspondant au coût le plus faible.

Le coût le plus élevé que l'on puisse obtenir avec ce schéma de codage correspond à une partition constituée uniquement de carrés de taille la plus petite possible, soit $2^l \times 2^l$, comme dans le cas du codage par arbre quaternaire, à cause des hypothèses faites sur l'apparition de chaque symbole. Le coût le moins élevé consiste aussi à avoir des carrés de taille 64×64 uniquement. En gardant la même notation qu'au paragraphe précédent, on a:

$$(6-l) \cdot 2^{2(q-6)} \leq C \leq 4(q-l-1) \quad (3.17)$$

Comme en général il y a beaucoup plus de carrés de taille $2^l \times 2^l$, il s'avère plus efficace que le codage par arbre quaternaire. Les exemples présentés à la section 3.5 confirmeront ceci dans le cas d'images réelles. La figure 3.11 illustre comment comptabiliser le nombre de bits nécessaires pour une certaine partition.

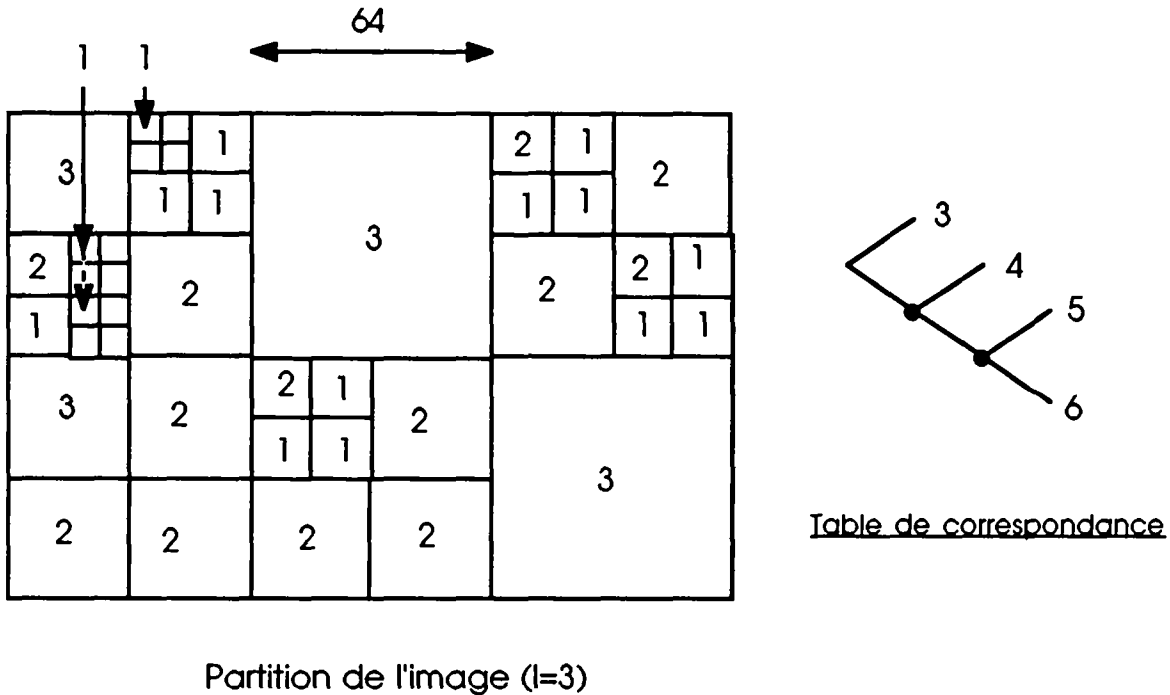


Figure 3.11: Le nombre de bits nécessaire à la transmission de chaque carré de la partition ci-dessus est indiqué au centre du carré.

Pour trouver le coût nécessaire à la représentation de la partition, on compte le nombre de carrés de chaque taille correspondant aux branches les plus à gauche à partir de chaque noeud de l'arbre quaternaire associé à la partition (attention, si l'on utilise un arbre quaternaire pour représenter la partition de l'image, le code conditionnel par taille de carré est utilisé ici). Notons ainsi G_1 à G_6 ces nombres pour chaque taille possible de carré. Les nombres de carrés restant sont notés similairement N_1 à N_6 . Le coût est alors donné par la relation:

$$C = (0 \times N_1 + 1 \times G_1) + (1 \times N_{1+1} + 2 \times G_{1+1}) + \dots + ((5-l) \times N_5 + (6-l) \times G_5) + (6-l) \times (N_6 + G_6)$$

$$= G_6 - \sum_{i=1}^6 ((i-1) \times (N_i + G_i) + G_i) \quad (3.18)$$

Disons encore que le code conditionnel présente l'avantage de pouvoir s'appliquer à des images rectangulaires de taille quelconque.

3.4.2 Quantification des coefficients polynomiaux:

Ce thème sera traité en détail au chapitre 5 après l'opération de rassemblement. Il s'agit de donner dans ce paragraphe quelques indications qui permettront d'estimer d'une manière indicative la compression que l'on peut atteindre déjà à ce niveau de la segmentation.

Quelle que soit la forme de l'approximation, on montre justement dans ce chapitre que les coefficients d'approximation des polynômes peuvent être quantifiés avec au plus huit bits par coefficient en conduisant à une erreur de quantification négligeable. On admet que l'image originale a été quantifiée à huit bits.

En effet, si l'on a r coefficients d'approximation, il suffit de transmettre r pixels quantifiés à huit bits du carré approximé. On est alors en mesure de calculer les autres pixels du carré en interpolant les r pixels précédents au moyen des mêmes fonctions de base $\Psi_j(x,y)$. La seule erreur que l'on peut commettre provient de l'arrondissement des r valeurs approximées au niveau de quantification le plus proche selon la méthode présentée à la section 2.3.3. On peut garder la même philosophie et décider de quantifier chaque coefficient avec moins de bits, soit b ce nombre. Il faut dans ce cas veiller à choisir de manière adéquate les r pixels quantifiés à b bits là où les erreurs d'arrondi sont les plus faibles. Cette approche sera utilisée dans la présentation des résultats de la section suivante.

On n'a cependant absolument pas tenu compte de la statistique des vecteurs de coefficients u sur l'ensemble des carrés de l'image. Or, il existe une redondance certaine de ces coefficients pour des carrés appartenant à un même espace régulier. Une approche pour éliminer cette redondance consiste à effectuer le rassemblement des carrés définissant un même espace régulier comme il est proposé au chapitre suivant. Si l'on ne désire pas changer la forme de la segmentation, on peut essayer de quantifier vectoriellement l'espace des vecteurs u . Ceci sort toutefois du cadre de la recherche effectuée ici.

On se contentera néanmoins de trouver un code simple permettant de réduire cette redondance dans le cas d'une approximation d'ordre zéro. Ceci permet d'avoir une idée de la compression atteignable sans autre après la procédure de rassemblement.

Au lieu de coder la valeur moyenne de chaque carré représentant l'approximation optimale au sens des moindres carrés au moyen de la fonction de base $\Psi_1(x,y)=1$, on va utiliser un code appelé CDC (Code de différence de carrés). Dans ce code, on transmet la différence entre les valeurs moyennes de deux carrés adjacents en supposant que l'on parcourt l'image selon une certaine convention

(par exemple, de gauche à droite puis de bas en haut). La longueur des mots codes associés à chaque différence possible a été établie sur la base de la statistique de ces différences. On a utilisé pour ce faire le code optimal de Huffman. Les résultats correspondants sont présentés à la section suivante.

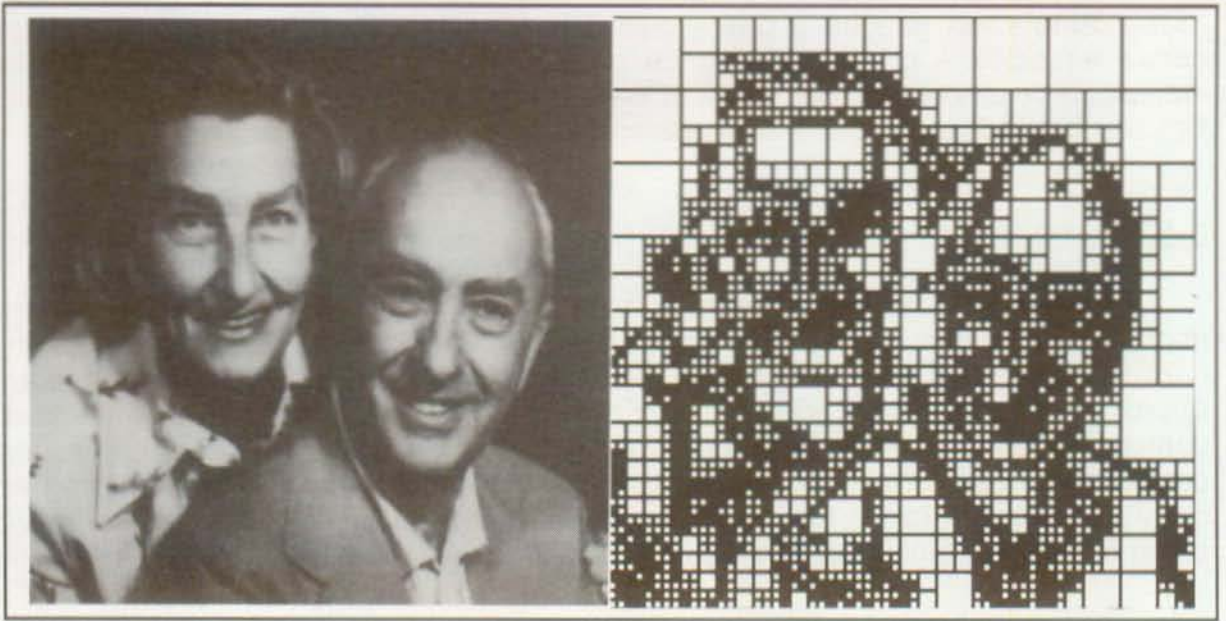
3.4.3 Postraitement:

En observant les images approximées, notamment si l'on tolère des erreurs importantes, on peut remarquer l'apparition de faux contours aux frontières de certains carrés. Ce défaut est lié au choix de fonctions polynomiales pour approximer les données de départ. En principe, l'attribution du statut de faux contour ne peut se faire que par la comparaison avec l'image de contrôle des contours. De tels artefacts ne peuvent cependant apparaître qu'à la frontière des carrés de grande taille. En effet, dans le cas de petits carrés, le polynôme va pratiquement interpoler les données de départ.

Pour éliminer ces faux contours correspondant à la présence d'une information haute fréquence locale, il suffit de filtrer passe-bas l'image dans le voisinage de ces sauts artificiels de luminance. Plus le carré sera grand, plus le filtrage devra être important et plus la zone où il sera effectué sera large. La largeur de la zone filtrée a été ainsi choisie à 2^{i-1} pixels pour un carré de taille $2^i \times 2^i$. Ce choix a été fait de manière rendre la dimension de la zone lissée une fonction croissante de la taille des carrés. Une taille plus faible que 2^{i-1} ne permet pas d'éliminer correctement les faux contours. Une taille plus grande risque de dégrader de façon trop importante le signal d'approximation polynomial estimé selon le critère des moindres carrés. Le filtre passe-bas est calculé de manière à minimiser les variations du gradient local estimé perpendiculairement au faux contour (voir section 5.3)

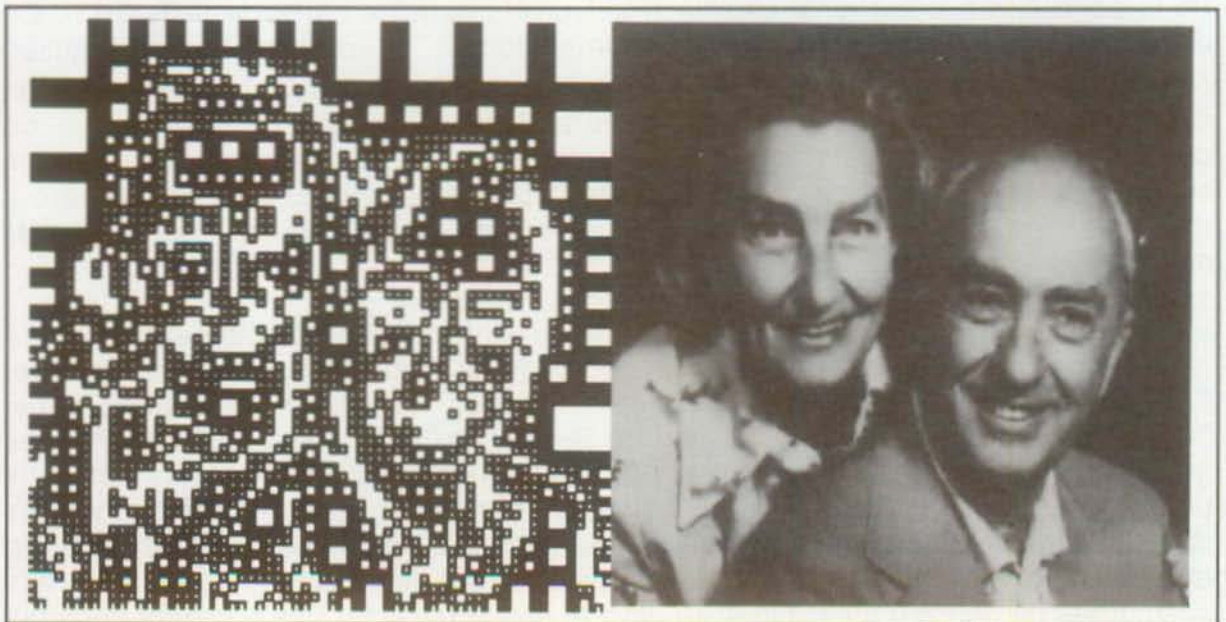
De manière à éviter de transmettre de l'information supplémentaire à un récepteur pour parvenir à déceler quels sont les faux contours, on admet de filtrer l'image dans le voisinage des frontières de tous les carrés de grande taille, l'apparition d'un contour réel se situant à la frontière exacte entre deux carrés de grande taille étant relativement peu probable. Pour ce qui est des frontières ne présentant pas de faux contours, l'effet du filtrage sera imperceptible vu ses caractéristiques basses fréquences. La taille minimale des carrés qui présentent de faux contours a été fixée 8×8 pour des polynômes de degré 2 ou 3, à 4×4 pour des polynômes de degré 0 ou 1. On évite ainsi le risque de lisser des contours réels qui ne peuvent apparaître que pour des carrés de dimension immédiatement inférieure aux tailles précédemment indiquées (voir algorithme de la figure 3.7).

La figure 3.12 présente l'image du couple rehaussée lorsqu'on utilise des plans pour approximer chaque carré. La figure 3.12a montre la forme de la partition (les carrés de taille 1×1 sont présentés en gris); la figure 3.12b montre l'image approximée; la figure 3.12c indique en noir les régions où est effectué le filtrage passe-bas de rehaussement; enfin la figure 3.12d montre l'image restaurée. Pour faire apparaître l'erreur maximum tolérée en chaque pixel, on exprime le seuil T_1 comme un nombre élevé au carré du fait de l'aspect quadratique de la mesure choisie. Cette remarque reste valable pour la présentation de tous les résultats expérimentaux de ce chapitre.



a) Image approximée

b) Partition de l'image (5290 carrés)



c) Zones filtrées passe-bas (en noir)

d) Image restaurée.

Figure 3.12: Postraitement de l'image du couple
(approximation par polynôme de degré 1(plan), $T_1=15^2$).

3.5 Analyse expérimentale de l'algorithme de division :

Dans cette section, nous allons mesurer la qualité des résultats de la segmentation en carrés après application de l'algorithme de division. Dans un premier temps, on essaiera d'appliquer cet algorithme sans l'utilisation d'une image de contrôle des contours selon le schéma de principe de la figure 3.2. Dans

le deuxième paragraphe, on discute des résultats lorsqu'on applique le schéma de l'algorithme simplifié de la figure 3.7, où l'on fait usage cette fois d'une image de contrôle des contours. On y relève en particulier l'étude du rôle du seuil T_1 sur la mesure d'erreur quadratique moyenne, du degré du polynôme d'approximation et de la compression d'information qui peut être obtenue à ce niveau du traitement. Enfin le dernier point analysé dans cette section traite de la dégradation qui apparaît si l'on limite volontairement la plus petite taille de carré à un niveau supérieur à celui indiqué par la relation (3.1).

Convention:

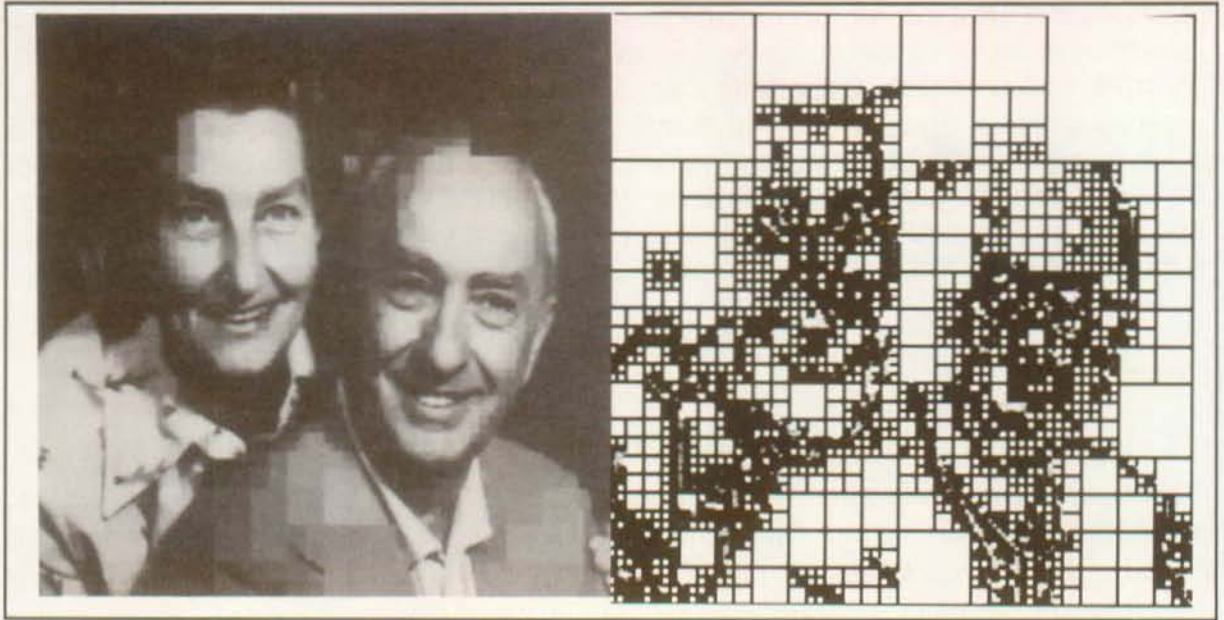
Les images présentées dans ce qui suit utilisent les notations suivantes. T_1 correspond au seuil sur l'erreur quadratique moyenne. α définit le degré du polynôme d'approximation. C_q et C_l indiquent les coûts nécessaires au codage de position des carrés lorsqu'on utilise le codage par arbre quaternaire et la codage conditionnel par taille de carré respectivement. ρ définit le rapport de compression sachant que l'on a codé chaque coefficient du polynôme avec 8 bits. Il s'exprime par la relation:

$$\rho = 256^2 \times 8 / (8rw + \min(C_q, C_l)) = 524288 / (8rw + \min(C_q, C_l)) \quad (3.19)$$

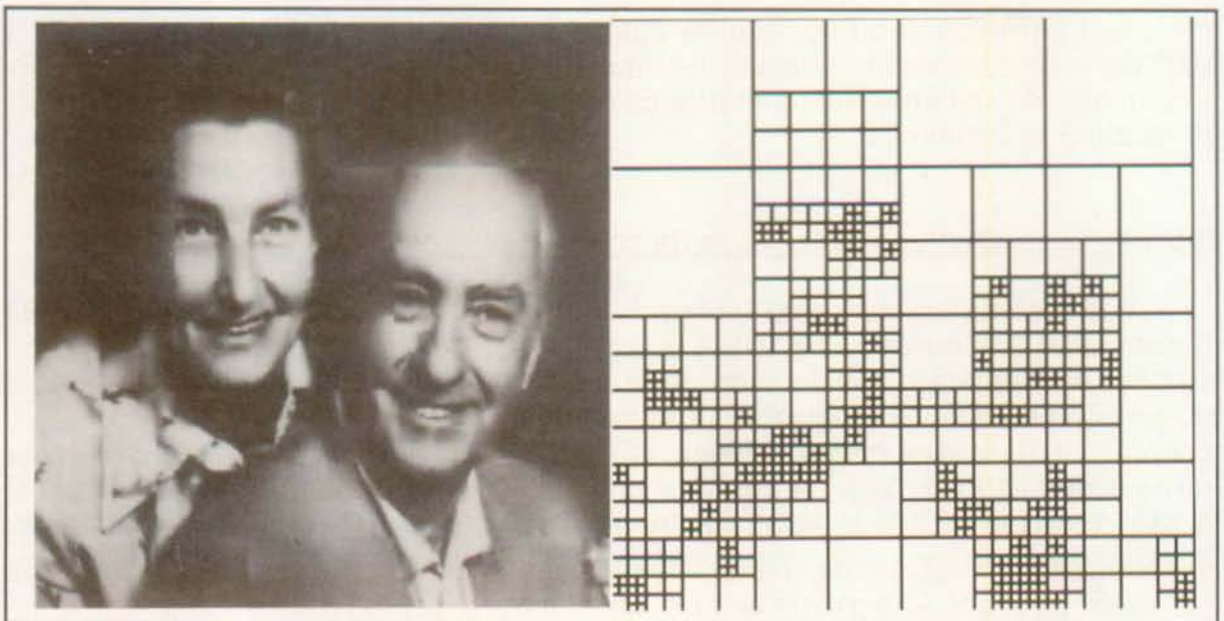
où r correspond au nombre de coefficients et w au nombre de carrés. Mis à part dans le paragraphe suivant, les images seront toujours divisées en utilisant une image de contrôle des contours calculée au moyen du filtre haute-fréquence présenté à la section 2.2.

3.5.1 Division effectuée sans image de contrôle des contours:

Les exemples qui sont présentés ici illustrent clairement la perte de l'information contour si l'on n'utilise qu'une mesure d'erreur quadratique moyenne comme seul critère d'appréciation de la qualité. Ceci reste indépendant du degré du polynôme choisi. La figure 3.13 montre les résultats obtenus pour l'image du couple. La première image indique la forme de la segmentation, la deuxième l'image reconstruite. Les degrés des polynômes utilisés dans les figures 3.13a, 3.13b, d'une part, 3.13c et 3.13d, d'autre part, sont 0 et 3 respectivement. Le seuil T_1 est fixé à 15^2 , le signal de départ étant quantifié à 255 niveaux disponibles. Les résultats peuvent être partiellement améliorés si l'on applique l'algorithme de rehaussement proposé au dernier paragraphe de la section précédente. Les images restaurées sont présentées à la figure 3.14. En gardant une même valeur de seuil pour T_1 , on présente à la figure 3.15 les résultats de l'algorithme de division pour l'image du bâtiment. Le degré du polynôme correspondant à 3.15a et 3.15b est fixé à 1 alors qu'il est de 2 pour 3.15c et 3.15d.

a) Image approximée ($\alpha=0, T_1=15^2$)b) Partition de l'image ($w=8317$).

$C_q = 6056$ bits $C_l = 6618$ bits $\rho = 7.22$.

c) Image approximée ($\alpha=3, T_1=15^2$)d) Partition de l'image ($w=583$).

$C_q = 464$ bits $C_l = 522$ bits $\rho = 11.13$.

Figure 3.13: Division de l'image du couple sans image de contrôle des contours.

Les images de la figure 3.13 mettent bien en évidence la perte de l'information contour que l'on a si l'on base le critère de division uniquement sur une mesure d'erreur quadratique moyenne. Ainsi les images des figures 3.13a et 3.13b font apparaître une perte des contours au voisinage des fronts des personnes ainsi que sur la veste de l'homme. Les carrés où l'on décèle ce genre de distorsion sont

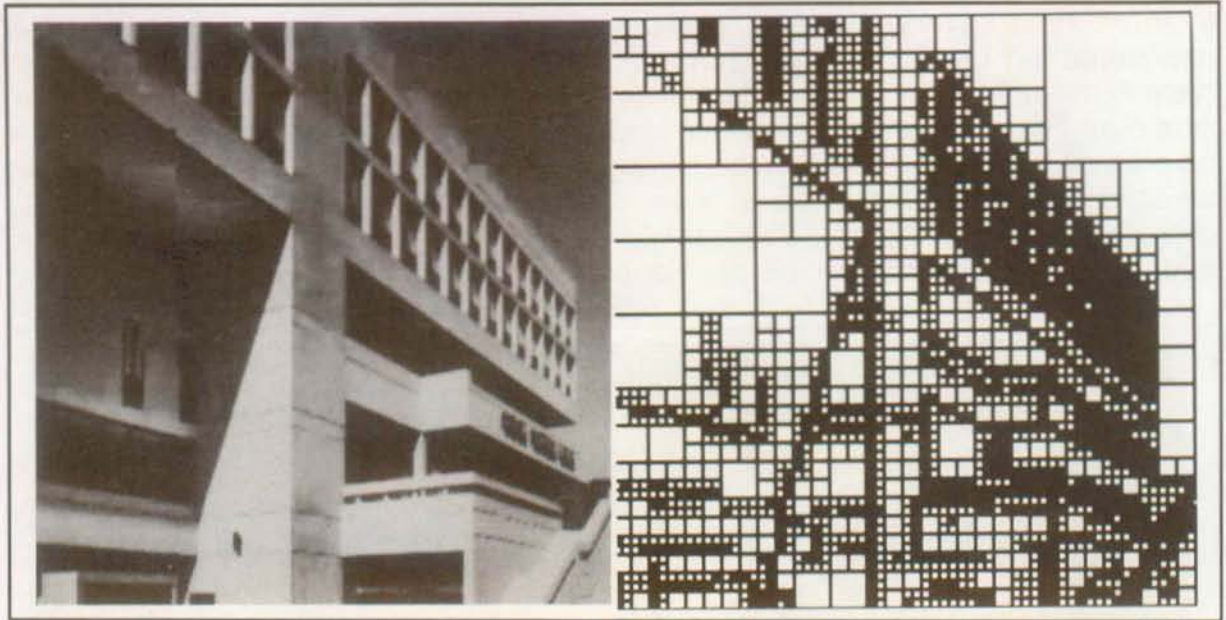
toujours de très grande taille ($64 \times 64, 32 \times 32$). En effet, si le signal est bien approximé sur une large proportion du carré, l'erreur quadratique correspondante reste relativement faible. La contribution à cette erreur correspondant à la région située de l'autre côté du contour est rendue négligeable par la pondération par le nombre de points du carré. Lorsqu'on augmente le degré du polynôme à 3 (figures 3.13c et 3.13d), les distorsions sont encore plus visibles au niveau de la perte des contours réels de l'image. Par contre, on réduit le nombre des faux contours apparaissant à la jonction entre des carrés adjacents.



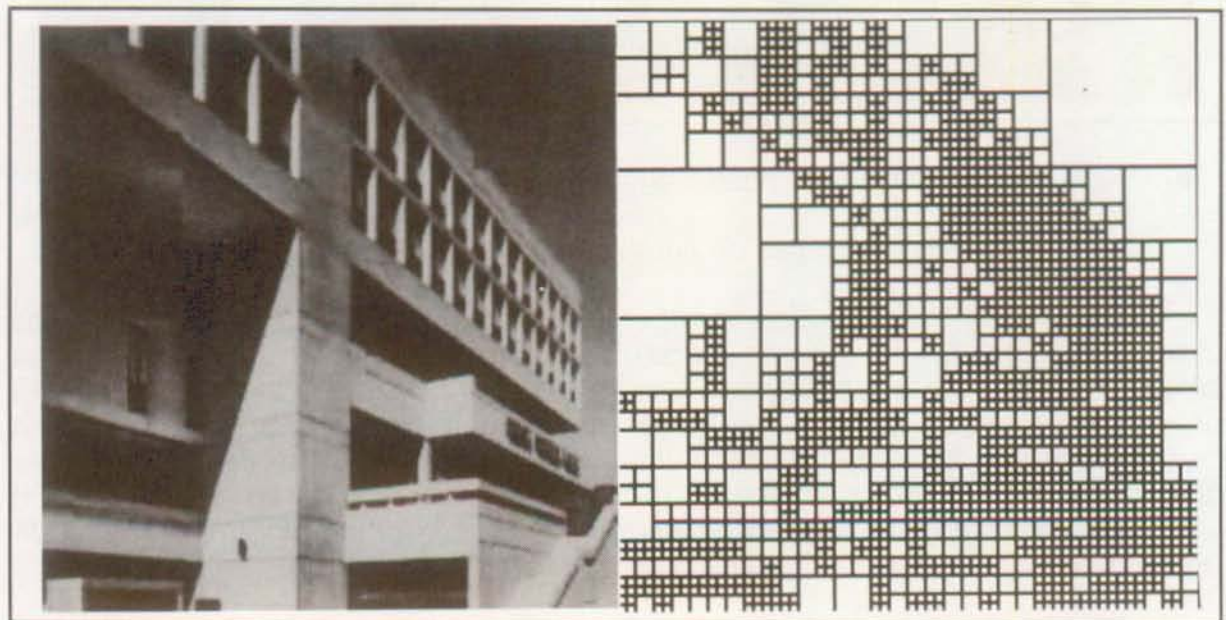
a) Image de la figure 3.13a rehaussée. b) Image de la figure 3.13c rehaussée

Figure 3.14: Images du couple de la figure 3.13 restaurées.

La figure 3.14 montre bien comment il est possible de réduire les distorsions dues à la perte des faux contours de l'image, en appliquant le filtre passe-bas sélectif par taille de carré proposé à la section 3.4.3. L'image de la figure 3.14b présente un aspect moins agréable à cause du faible nombre de carré de la partition obtenue au moyen de polynômes d'ordre 3. Il est important de remarquer que si ces dégradations peuvent être éliminées à ce niveau de la segmentation, il n'est plus possible d'obtenir les frontières réelles des contours après l'opération de rassemblement en partant d'un GCR basé sur les images segmentées de la figure 3.13.



a) Image approximée ($\alpha=1, T_1=15^2$) b) Partition de l'image ($w=4957$).
 $C_q = 2748$ bits $C_l = 2662$ bits $\rho = 4.31$.



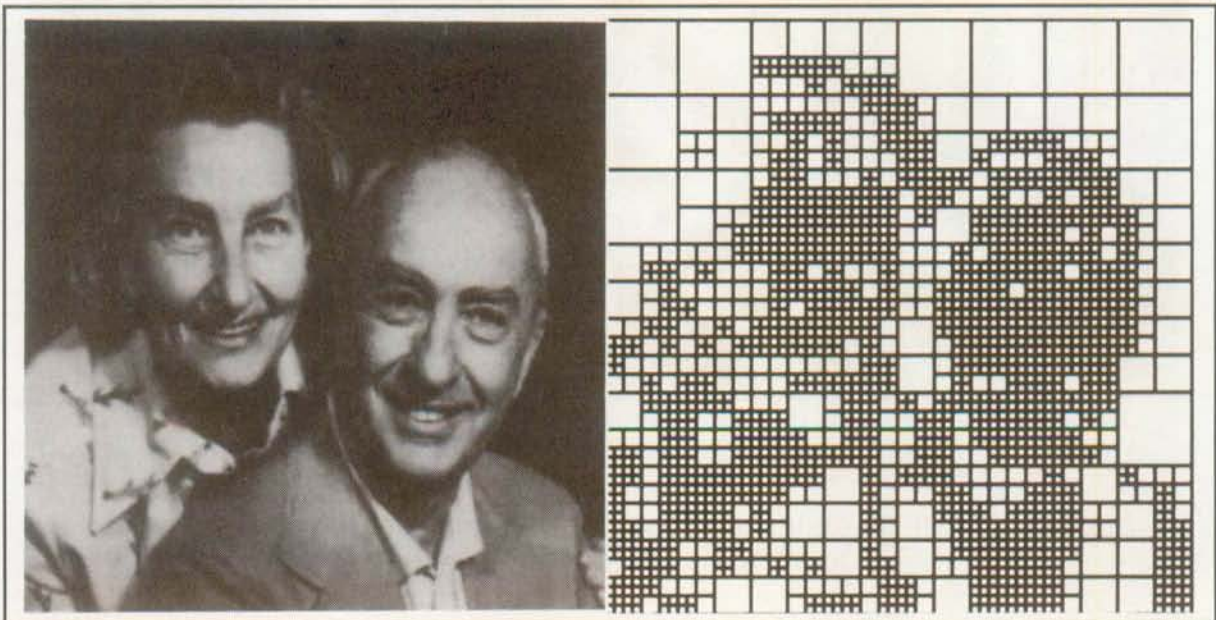
c) Image approximée ($\alpha=2, T_1=15^2$) d) Partition de l'image ($w=1825$).
 $C_q = 940$ bits. $C_l = 841$ bits $\rho = 5.93$.

Figure 3.15: Division de l'image du bâtiment sans image de contrôle des contours.

Les considérations effectuées à partir de la figure 3.13 peuvent être confirmées en observant les résultats de la figure 3.15. On remarque en plus que la perte de l'information contour est indépendante du degré du polynôme d'approximation. Disons encore que ce genre de dégradation apparaît presque toujours au voisinage des grandes régions homogènes de l'image (ciel pour

l'image du bâtiment, fond pour l'image du couple). Ceci résulte du choix du critère d'erreur globale. En gardant un même nombre de régions pour une même image, le critère d'erreur quadratique (EQ) aurait fait disparaître des régions homogènes de petite taille. Il aurait dès lors été impossible de rehausser la qualité de l'image par filtrage passe-bas local comme indiqué précédemment.

On peut éviter cependant de perdre l'information contour en plaçant le seuil T_1 beaucoup plus bas. Néanmoins, il en résulte une subdivision excessive de l'image de départ, certains carrés correspondant à des parties d'espace régulier ayant été découpés outre mesure. L'algorithme de rassemblement aura dès lors la tâche d'autant plus difficile. La figure 3.16 présente la division de l'image du couple lorsqu'on a choisi une approximation par un polynôme de degré 3 et une valeur de T_1 à 5^2 , si l'on a une quantification à 8 bits du signal original. L'image qui en résulte est presque parfaite si ce n'est que le nombre de régions de la partition est très élevé, ce qui réduit considérablement la compression d'information.



a) Image approximée ($\alpha=3, T_1=5^2$)

b) Partition de l'image ($w=2020$).

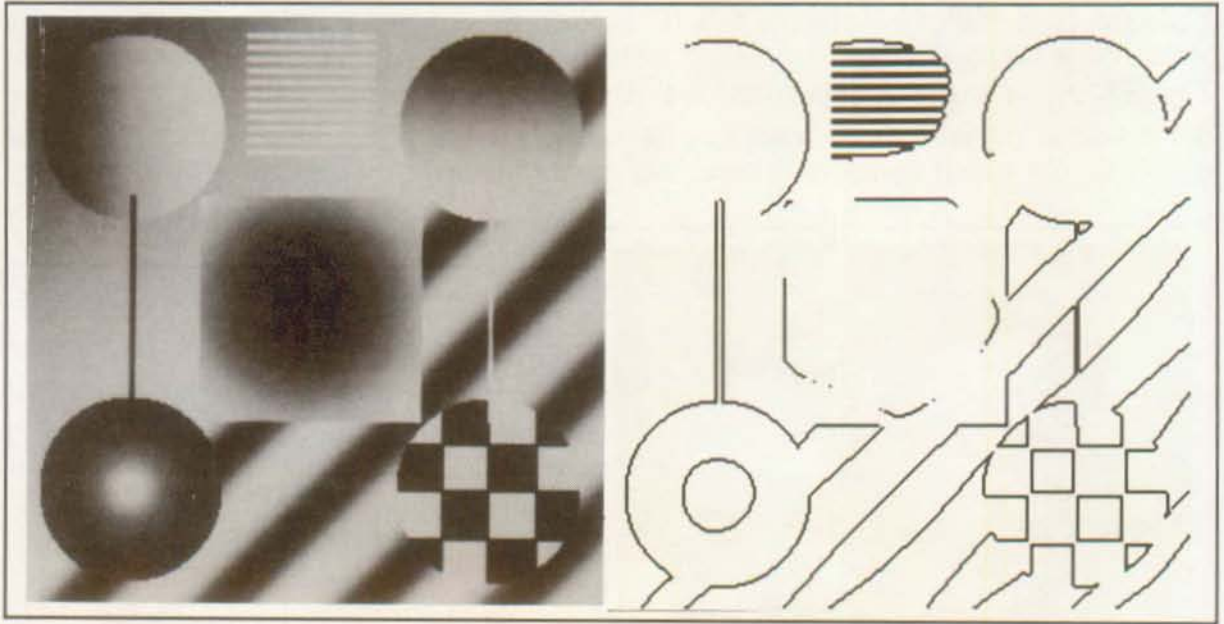
$$C_q = 976 \text{ bits} \quad C_l = 878 \text{ bits} \quad \rho = 3.23.$$

Figure 3.16: Division de l'image du couple sans image de contrôle des contours avec seuil sur l'erreur quadratique moyenne placé très bas.

De manière à éviter la perte de l'information contour sans toutefois subdiviser excessivement l'image, il suffit d'appliquer l'algorithme proposé à la figure 3.7 avec utilisation d'une image de contrôle des contours. Les résultats correspondants vont être présentés dans ce qui suit.

3.5.2 Effets des paramètres et compression:

Pour pouvoir estimer qualitativement les effets des paramètres de l'algorithme de division, ce dernier a été appliqué dans un premier temps à l'image de test de la figure 3.17a. La figure 3.17b définit l'image de contrôle des contours correspondante estimée au moyen du filtre haute fréquence de la section 2.2. Le tableau 3.18 indique les sous-numéros de la figure 3.19 pour les différentes valeurs du seuil et du degré du polynôme d'approximation dans l'application de l'algorithme de division à l'image de test.



a) Image de test (original)

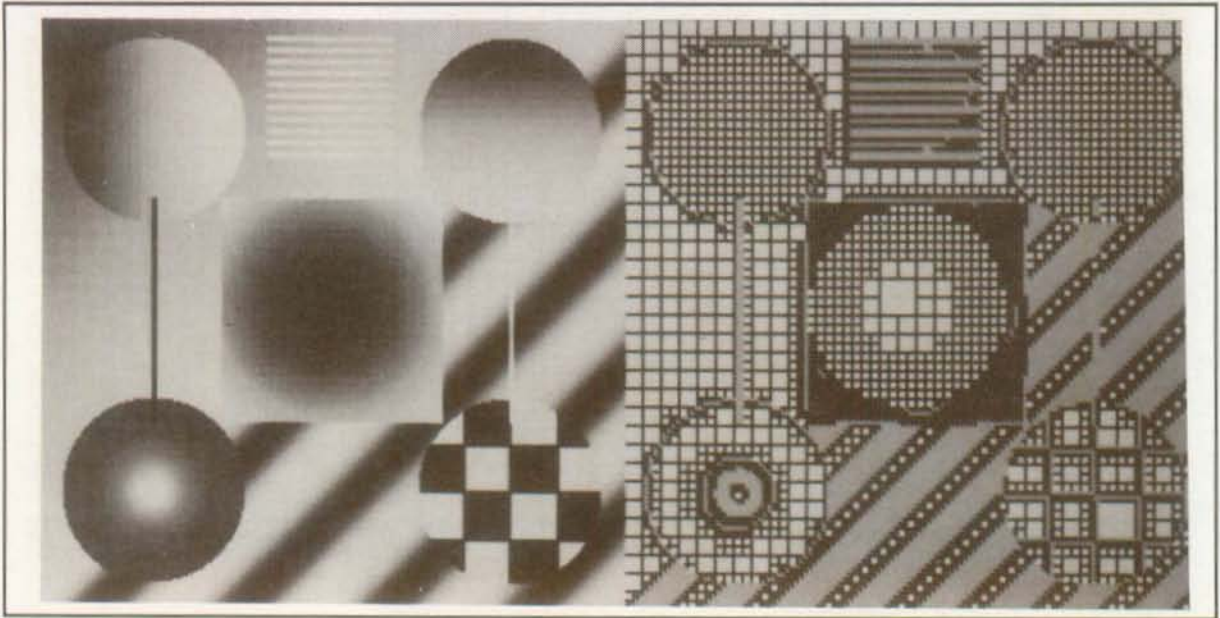
b) Image de contrôle des contours.

Figure 3.17: Image de test utilisée pour estimer l'effet des paramètres de l'algorithme de division de la figure 3.7.

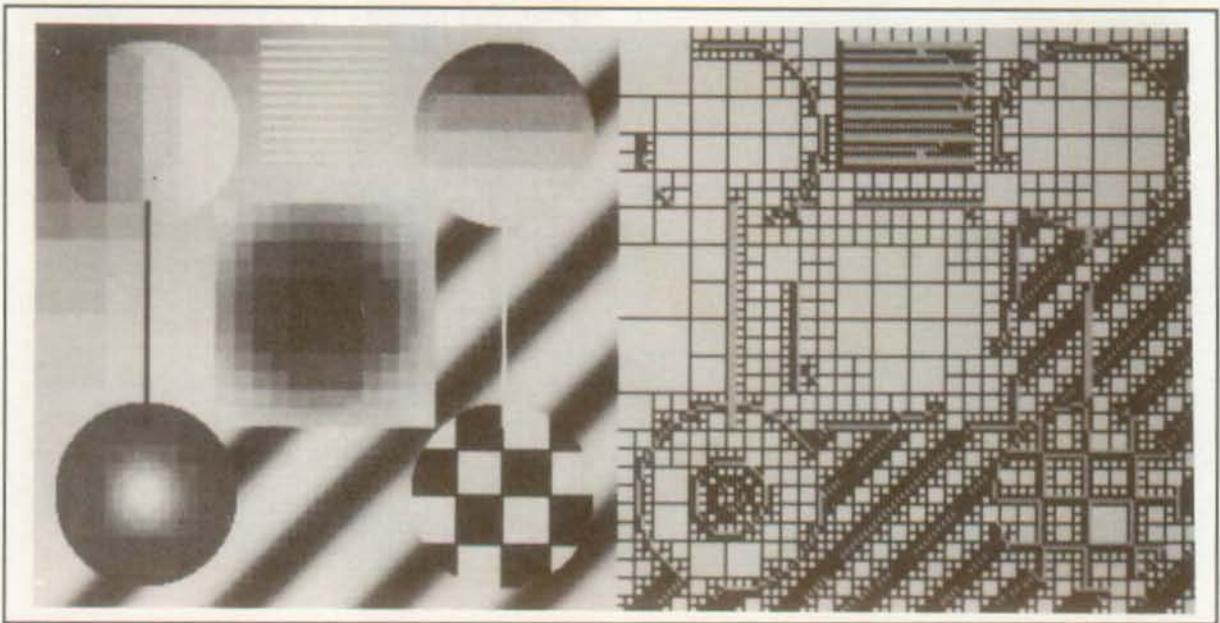
$\alpha \backslash T_1$	5^2	15^2	30^2
0	a	b	c
1	d	e	f
2	g	h	i
3	j	k	l

Tableau 3.18:

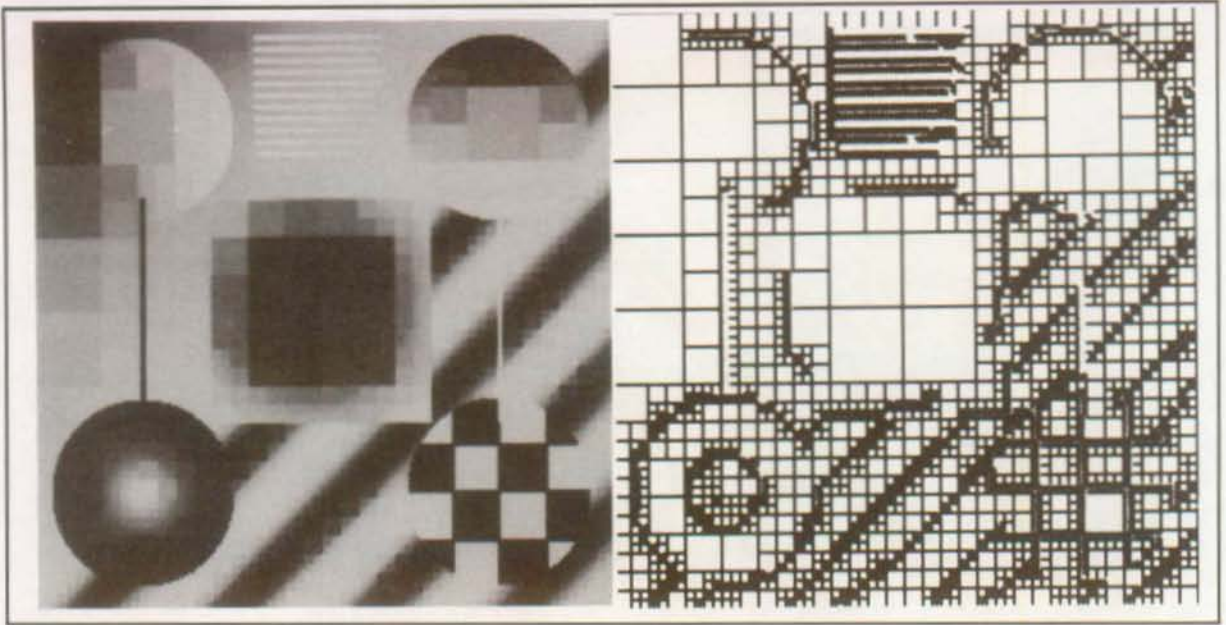
Numéros de la figure 3.19 associés à différentes valeurs des paramètres de seuil et de degré du polynôme.



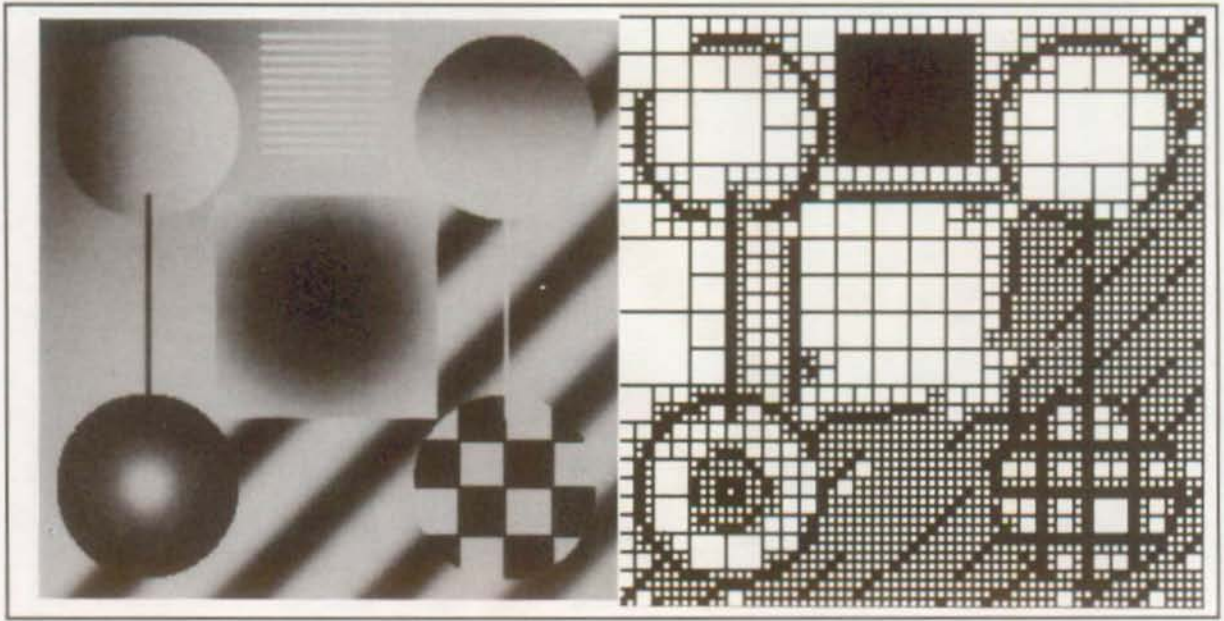
a) Image approximée ($\alpha=0, T_1=5^2$) et Partition de l'image ($w=21010$).
 $C_q = 11664$ bits $C_l = 11757$ bits $\rho = 2.92$.



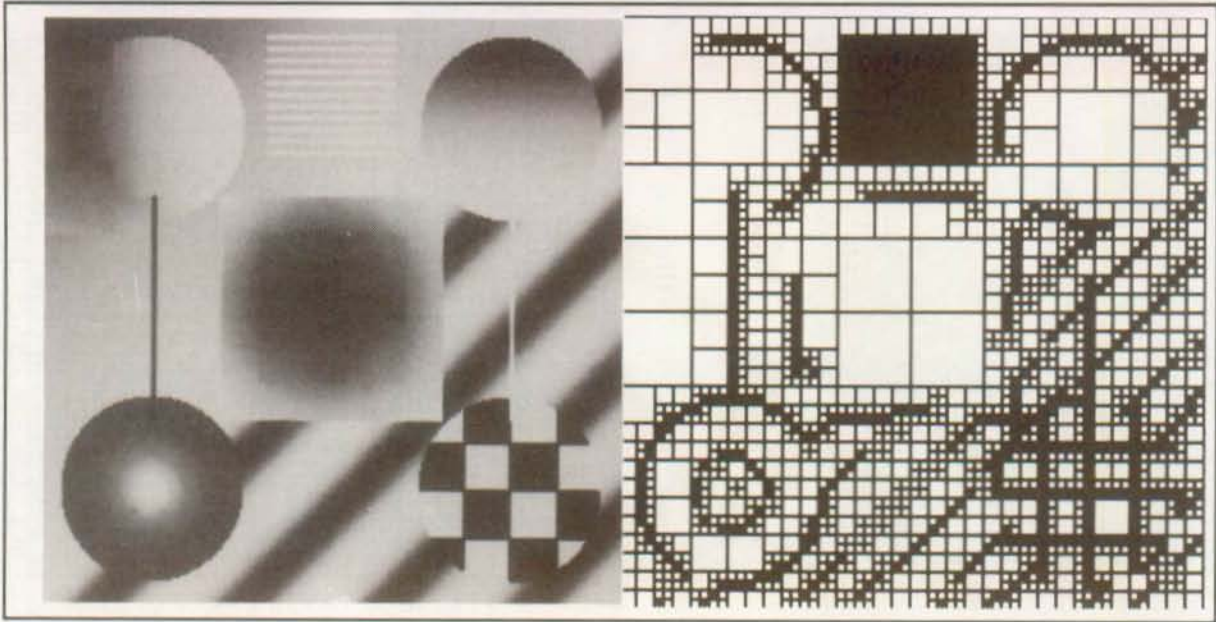
b) Image approximée ($\alpha=0, T_1=15^2$) et Partition de l'image ($w=9979$).
 $C_q = 7336$ bits $C_l = 8058$ bits $\rho = 6.01$.



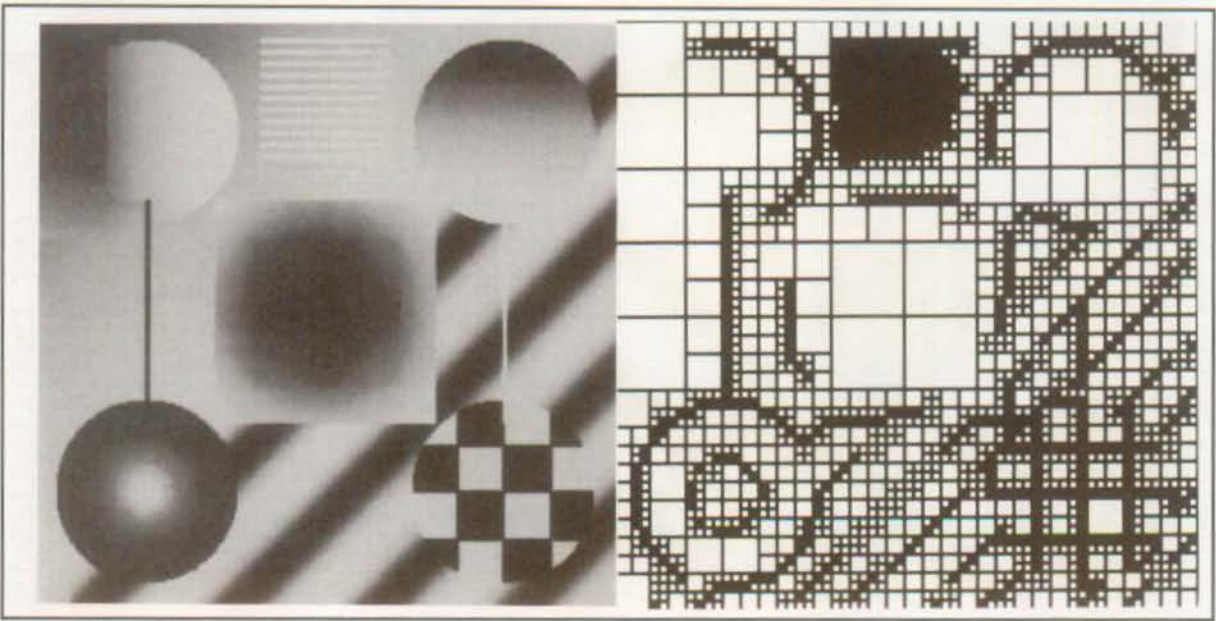
c) Image approximée ($\alpha=0, T_1=30^2$) et Partition de l'image ($w=8344$).
 $C_q = 5604$ bits $C_l = 6481$ bits $\rho = 7.25$.



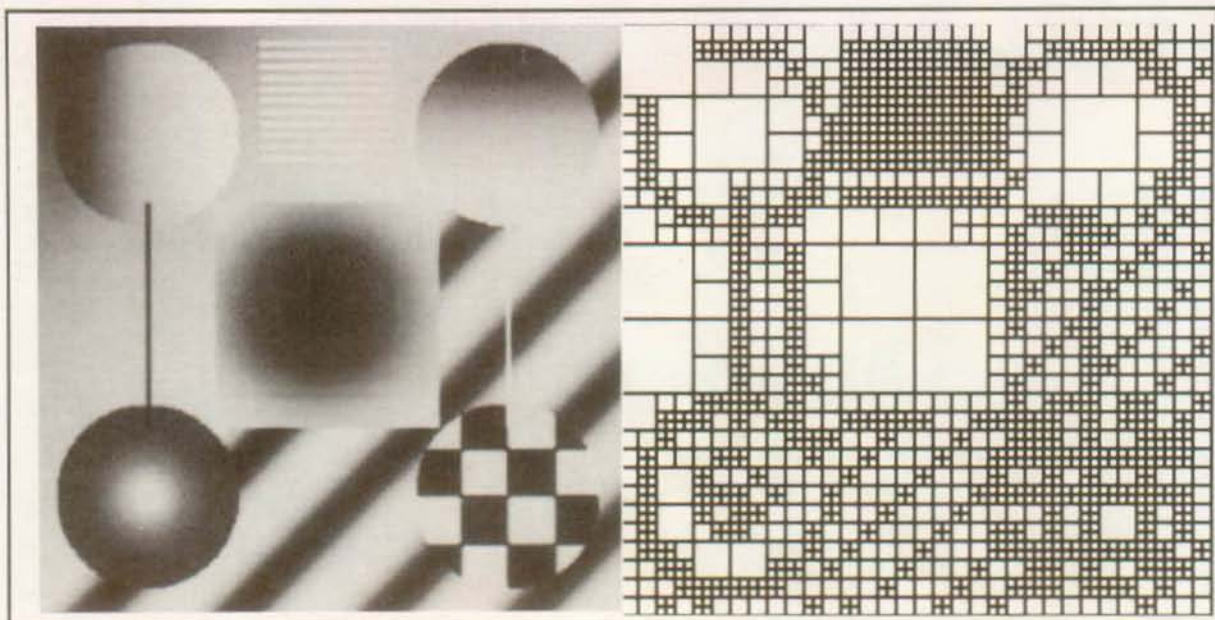
d) Image approximée ($\alpha=1, T_1=5^2$) et Partition de l'image ($w=4711$).
 $C_q = 3208$ bits $C_l = 3170$ bits $\rho = 4.51$.



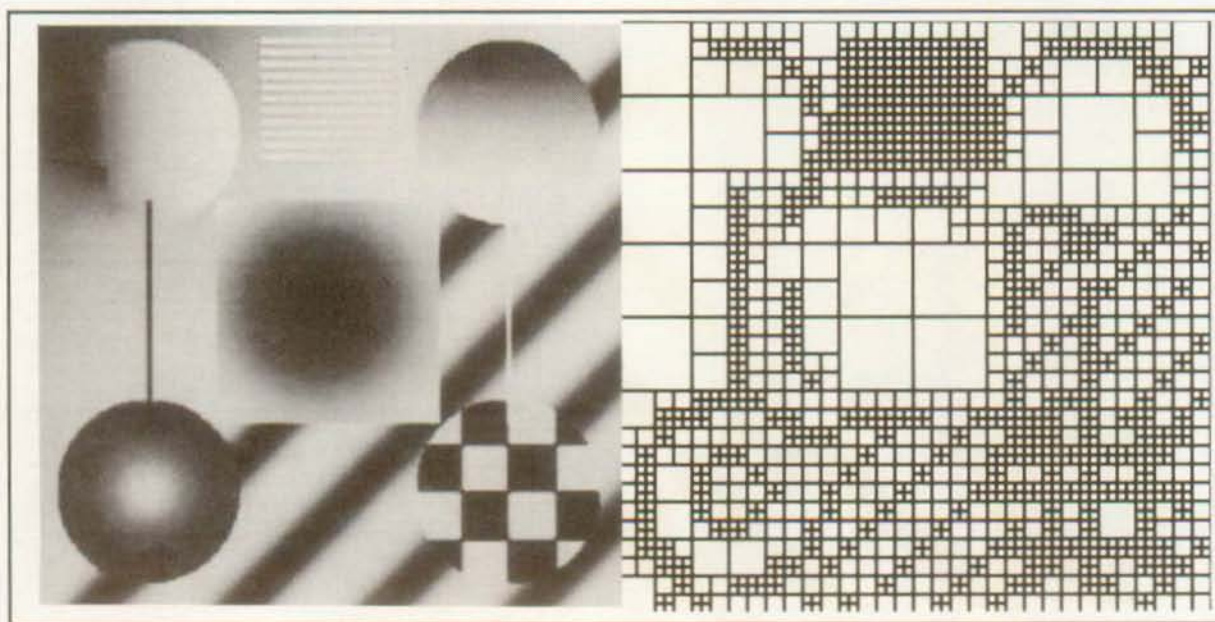
e) Image approximée ($\alpha=1, T_1=15^2$) et Partition de l'image ($w=3898$).
 $C_q = 2492$ bits $C_l = 2628$ bits $\rho = 5.46$.



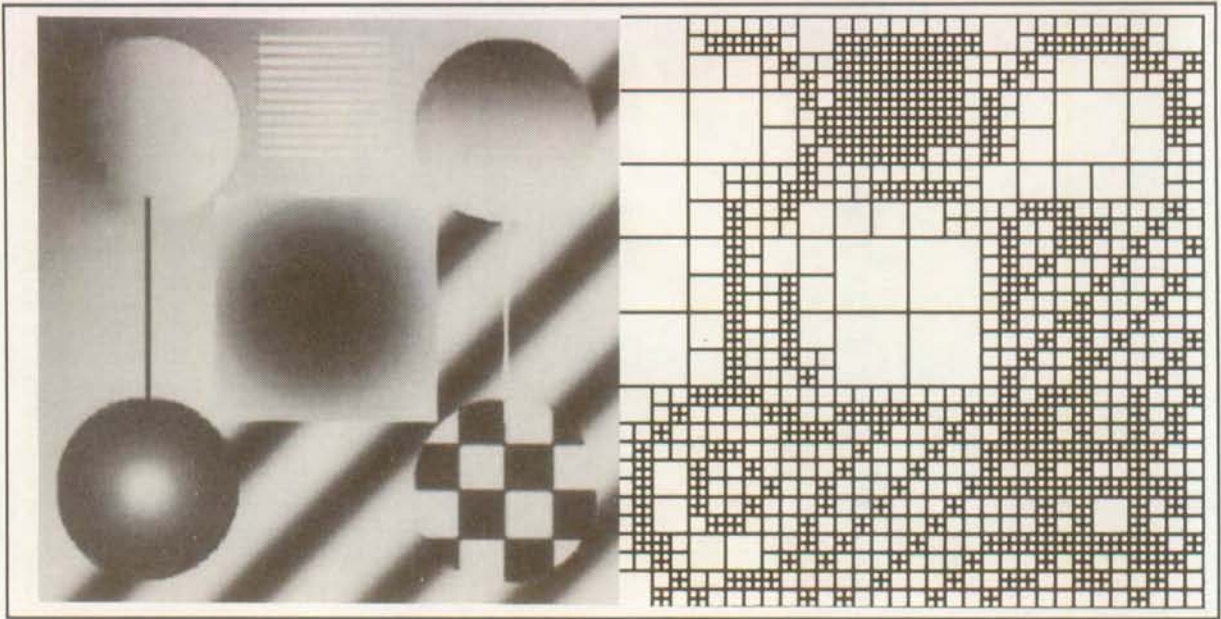
f) Image approximée ($\alpha=1, T_1=30^2$) et Partition de l'image ($w=3679$).
 $C_q = 2432$ bits $C_l = 2586$ bits $\rho = 5.78$.



g) Image approximée ($\alpha=2, T_1=5^2$) et Partition de l'image ($w=1813$).
 $C_q = 1036$ bits $C_l = 951$ bits $\rho = 5.90$.

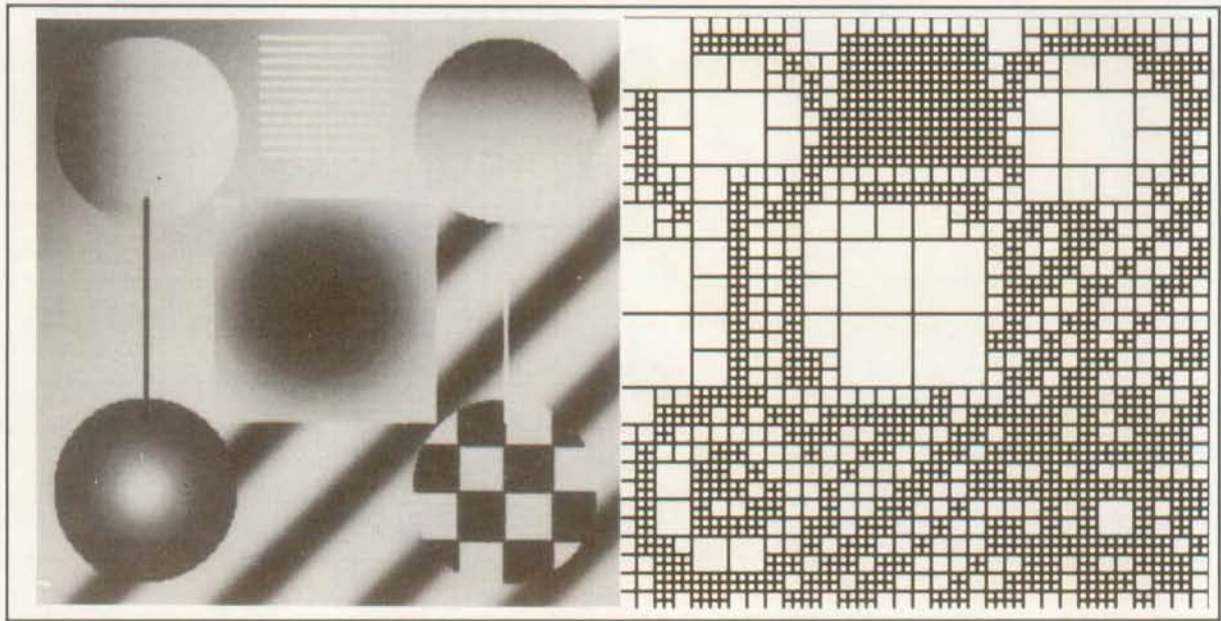


h) Image approximée ($\alpha=2, T_1=15^2$) et Partition de l'image ($w=1654$).
 $C_q = 984$ bits $C_l = 918$ bits $\rho = 6.53$.



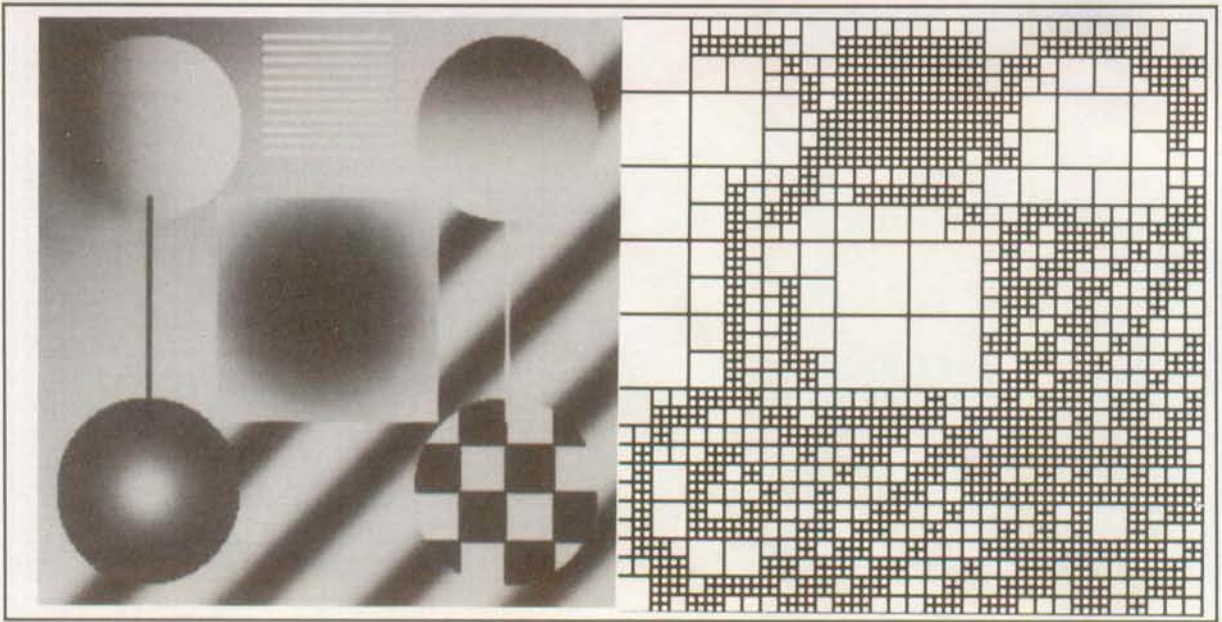
i) Image approximée ($\alpha=2, T_1=30^2$) et Partition de l'image ($w=1594$).

$$C_q = 984 \text{ bits} \quad C_l = 918 \text{ bits} \quad \rho = 6.77.$$



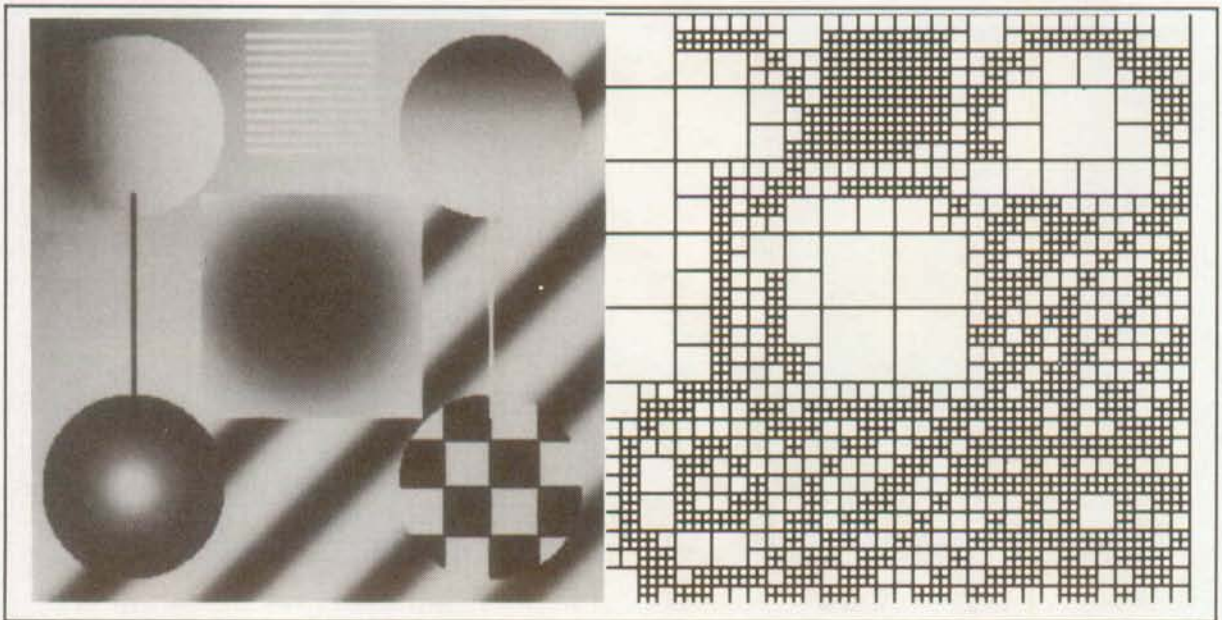
j) Image approximée ($\alpha=3, T_1=5^2$) et Partition de l'image ($w=1954$).

$$C_q = 1032 \text{ bits} \quad C_l = 945 \text{ bits} \quad \rho = 3.33.$$



k) Image approximée ($\alpha=3, T_1=15^2$) et Partition de l'image ($w=1849$).

$C_q = 984$ bits $C_l = 911$ bits $\rho = 3.52$.



l) Image approximée ($\alpha=3, T_1=30^2$) et Partition de l'image ($w=1819$).

$C_q = 984$ bits $C_l = 911$ bits $\rho = 3.58$.

Figure 3.19: Division de l'image de test avec différentes valeurs des paramètres.

Si l'image de contrôle des contours reproduit presque tous les contours, dans l'image reconstruite, ils restent préservés indépendamment de la valeur choisie pour T_1 .

Par contre, le degré du polynôme aura de l'influence sur la création de contours artificiels, si T_1 est trop élevé. En effet, plus la taille du carré est grande par rapport au nombre r des coefficients du polynôme, moins il est facile à la fonction d'approximation de suivre les variations du signal original au voisinage des frontières du carré. Bien entendu, ceci n'est pas valable si l'on se trouve dans un carré où la modification des paramètres u_i selon le critère des moindres carrés entraîne presque une interpolation des données de départ. Ce défaut peut être éliminé par l'application de l'algorithme de rehaussement décrit précédemment.

Enfin, on voit apparaître certains artefacts pour les carrés de plus petite taille contenant des contours importants si $r < 2 \times 2^l$, une interpolation des données de départ étant impossible. Les variations rapides entraînent des fluctuations incontrôlées de l'approximation polynomiale. En garantissant une correspondance entre frontières de régions et contours réels de l'image, la segmentation finale permettra d'éliminer ce genre d'erreurs.

Pour ce qui est de la forme de la segmentation, le nombre de carrés décroît avec l'augmentation du degré de l'approximation. La différence provient essentiellement des carrés de plus petite taille. En effet, la plupart des carrés apparaissent au voisinage des contours de l'image, à cause du critère de division choisi. Ceux-ci sont toujours de la plus petite taille possible (1x1 pour le degré 0, 2x2 pour le degré 1, 4x4 pour les degrés 2 et 3). Il existe donc pratiquement 4 et 16 fois plus de carrés de plus petite taille pour le polynôme de degré 1 et 0 que pour ceux de degrés 1 et 2, respectivement. Si reconstruire la partition est relativement plus coûteux sur le plan de la compression pour les polynômes de degré faible, il n'en va pas de même pour la mémorisation des coefficients d'approximation, leur nombre augmentant avec le degré du polynôme. On peut choisir de mémoriser seulement un certain nombre de coefficients, encore faut-il indiquer lesquels on a choisi et selon quel critère.

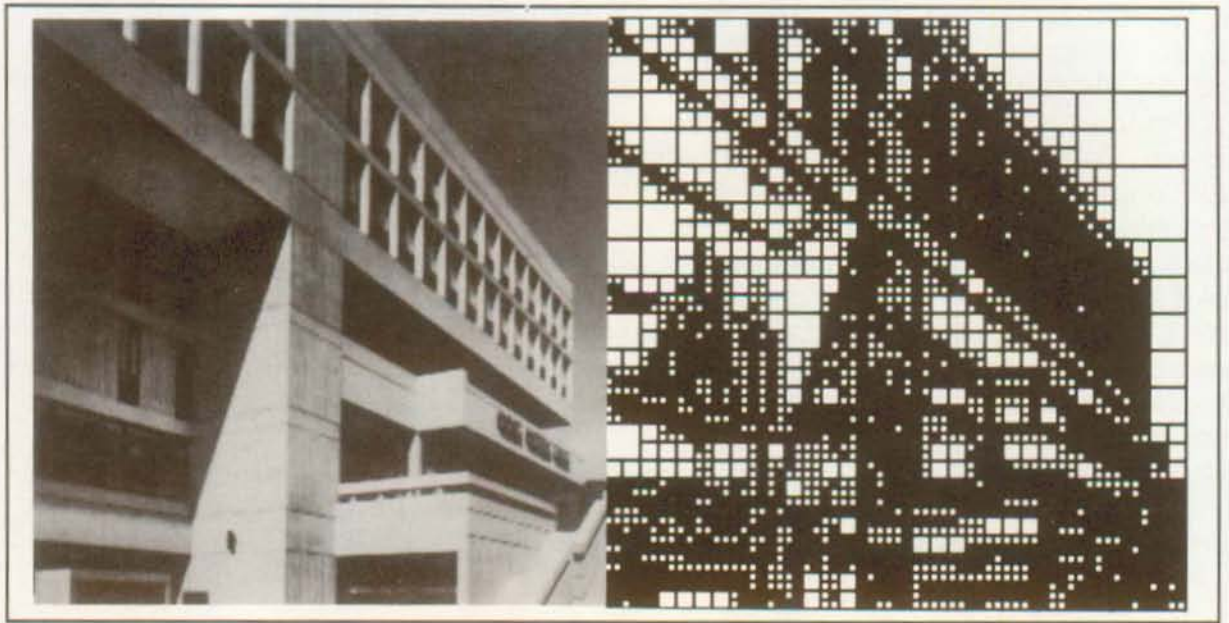
Ces différentes interprétations seront confirmées par des exemples sur les images naturelles des figures 2.2 à 2.4.

Rôle du seuil sur l'erreur quadratique moyenne:

De ce qui précède, il résulte que les contours existants de l'image sont préservés grâce à l'utilisation de l'image de contrôle. Le seuil T_1 permet de limiter le nombre de carrés ne présentant pas de contours tout en garantissant une bonne correspondance moyenne entre le polynôme approximant et le signal à l'intérieur de chaque carré. Ceci permettra d'accélérer le fonctionnement de l'algorithme de rassemblement, un moins grand nombre de régions devant être prises en considération. Néanmoins, il ne faut pas placer ce seuil à un niveau trop élevé, sans quoi on risquerait de ne pas avoir la meilleure correspondance entre fonctions polynomiales et les données originales après segmentation. En effet, lors de la procédure adaptative de rassemblement, on cherche toujours le lieu de dissimilarité minimum du graphe de contiguïté de régions pour atteindre un

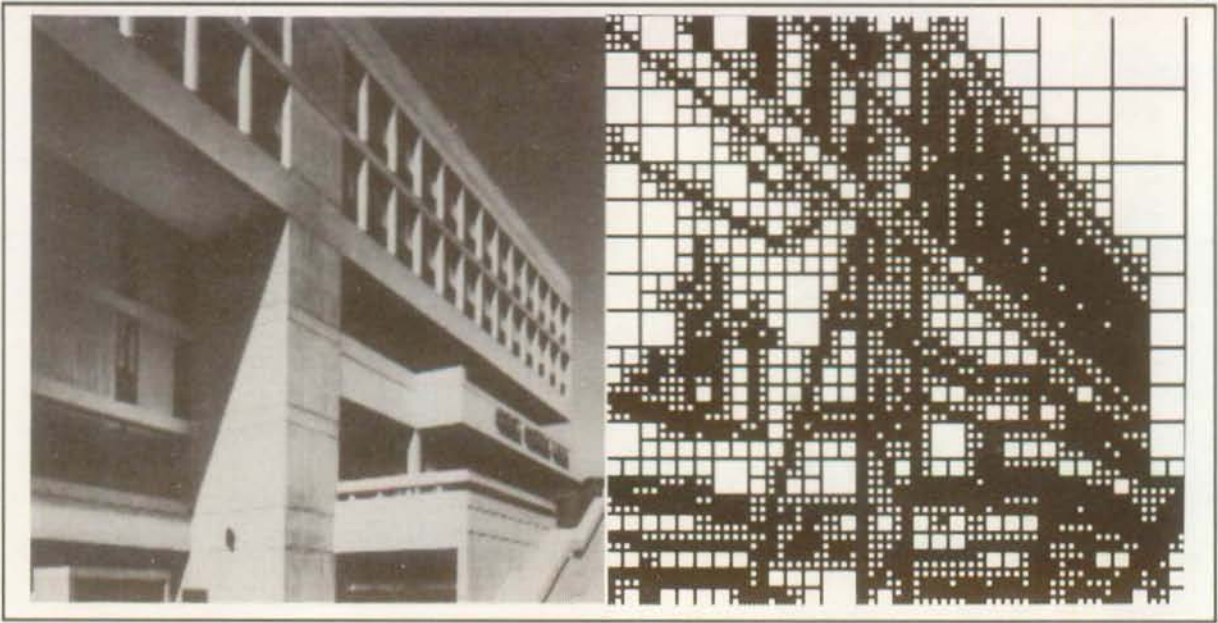
optimum. Celui-ci serait faussé par la structure initiale du graphe imposée par l'algorithme de division.

La figure 3.20 présente les résultats de l'algorithme de division appliqué au bâtiment pour trois valeurs de seuil (5^2 , 15^2 , 30^2) et une fonction polynômiale séparable de degré 2 définie par la relation (3.9) avec $r_1=r_2=2$. La figure 3.21 illustre la partition et l'approximation que l'on obtient de l'image du caméraman après application de l'algorithme de division en utilisant des polynômes de degré 2 et les mêmes valeurs pour le seuil T_1 que dans la figure précédente. En observant les figures 3.20 et 3.21, on remarque que la qualité des résultats reste inchangée quelle que soit la valeur de T_1 . En effet, à ce niveau de la segmentation les polynômes sont parfaitement à même de suivre des régions homogènes à variation lente. En comparant les deux figures, on ne remarque pas de dégradation au voisinage des contours de la figure 3.20, le polynôme choisi interpolant parfaitement les données images pour tous les carrés de taille 2×2 . Dans le cas de la figure 3.21, le polynôme est en mesure de suivre les variations brusques du signal luminance à l'emplacement des contours, mais des effets de bord sont visibles. En effet, dans ce cas, l'approximation au sens des moindres carrés ne peut interpoler les pixels du carré 4×4 (6 coefficients d'approximation contre 16 pixels).

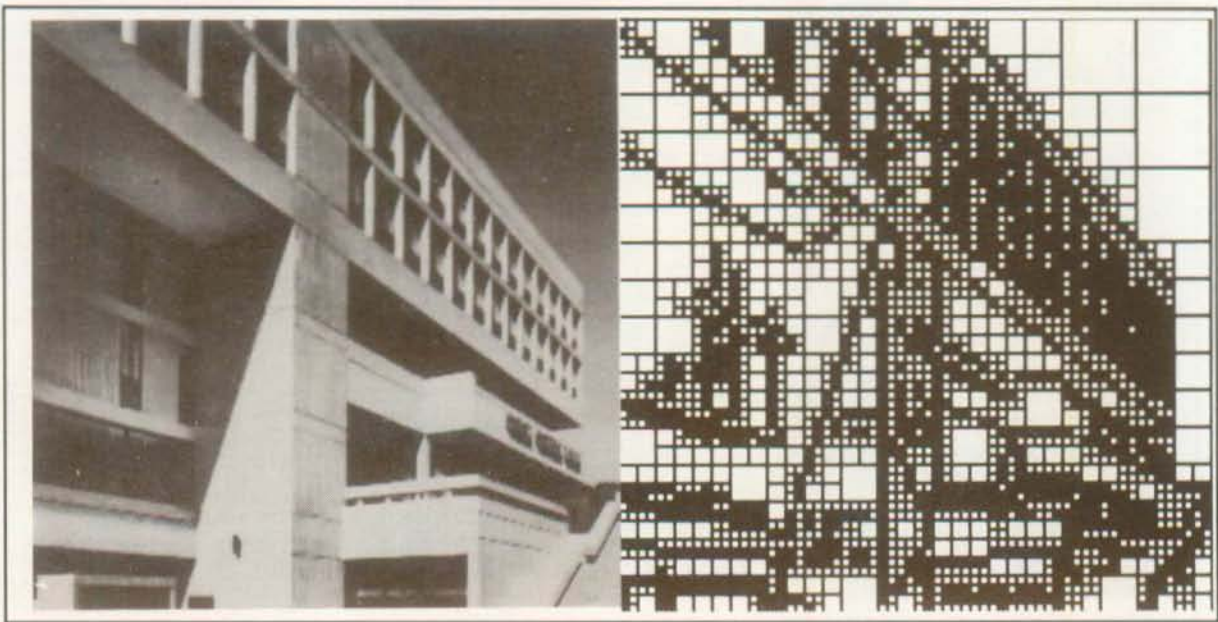


a) Image approximée ($r_1=2, r_2=2, T_1=5^2$) et Partition de l'image ($w=8749$).

$$C_q = 3996 \text{ bits} \quad C_l = 3477 \text{ bits} \quad \rho = 1.85.$$

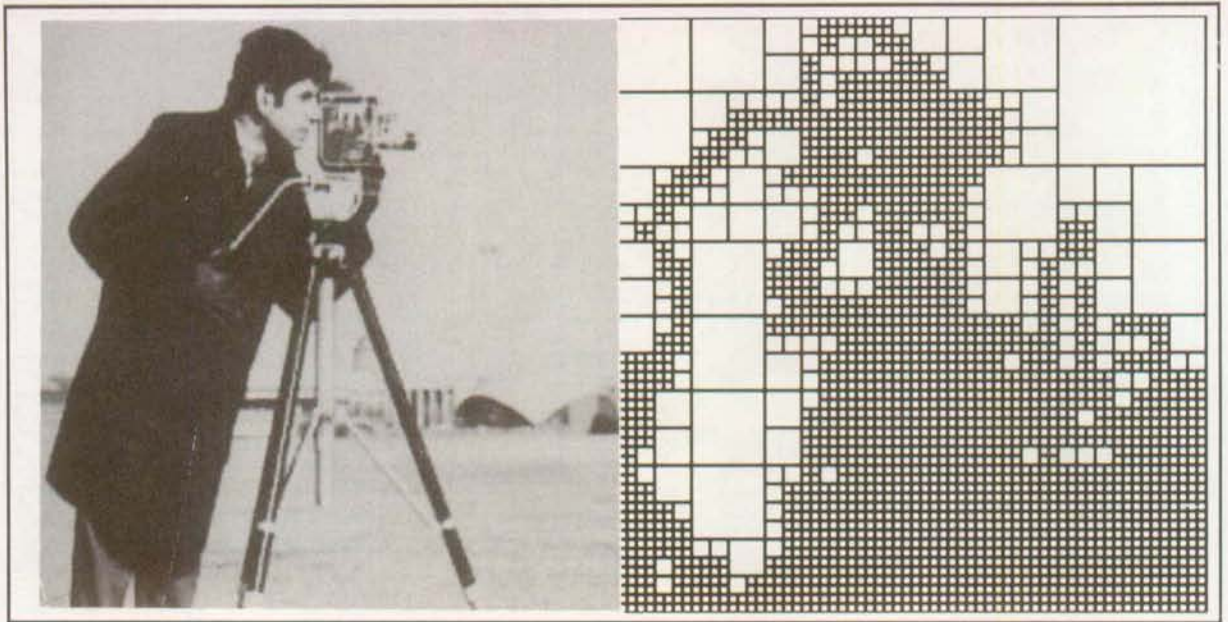


b) Image approximée ($r_1=2, r_2=2, T_1=15^2$) et Partition de l'image ($w=6847$).
 $C_q = 3608$ bits $C_l = 3302$ bits $\rho = 2.36$.

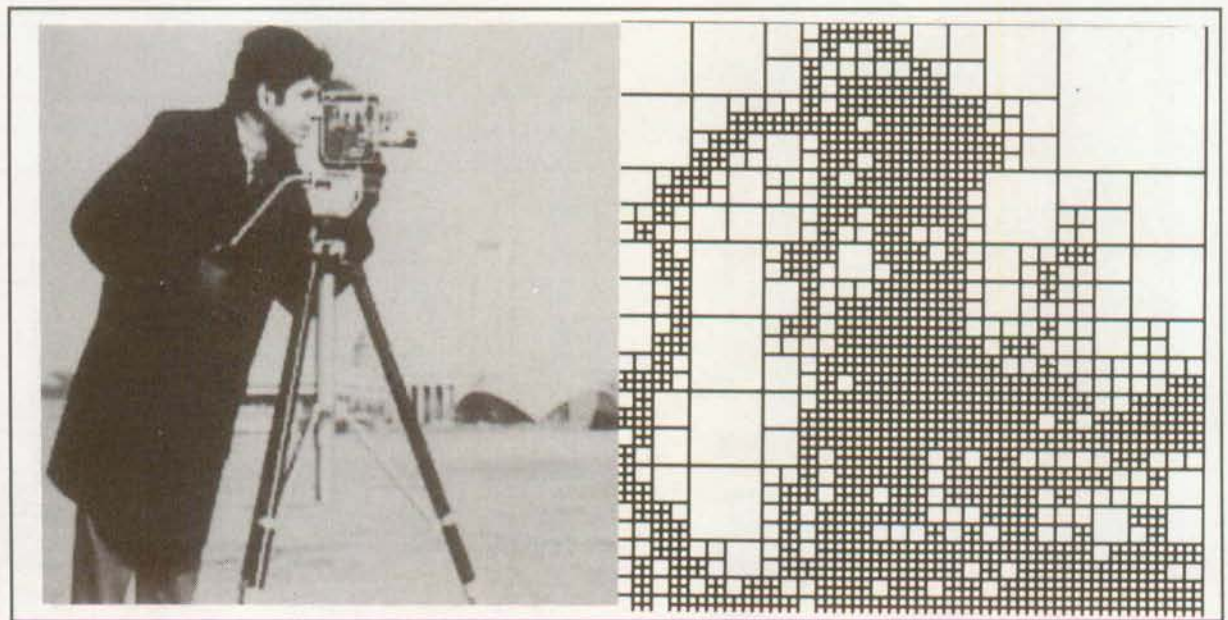


c) Image approximée ($r_1=2, r_2=2, T_1=30^2$) et Partition de l'image ($w=6391$).
 $C_q = 3544$ bits $C_l = 3294$ bits $\rho = 2.52$.

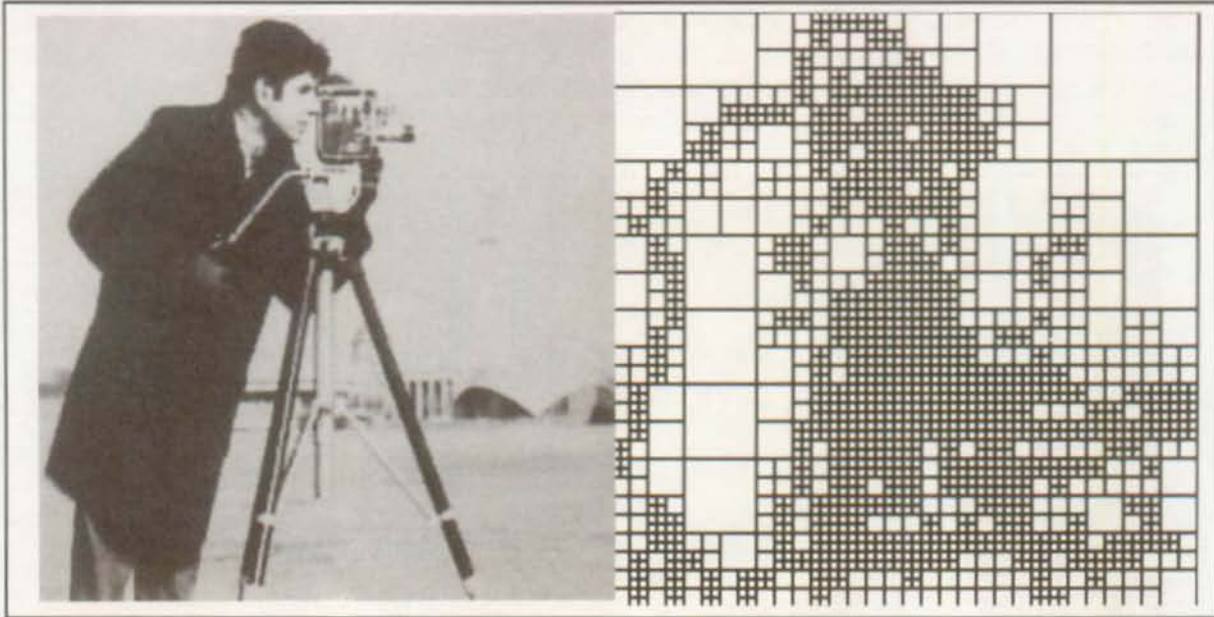
Figure 3.20: Division de l'image du bâtiment pour trois valeurs différentes de seuil sur l'erreur quadratique moyenne et une fonction polynomiale séparable.



a) Image approximée ($\alpha=2, T_1=5^2$) et Partition de l'image ($w=2293$).
 $C_q = 924$ bits $C_l = 799$ bits $\rho = 4.73$.



b) Image approximée ($\alpha=2, T_1=15^2$) et Partition de l'image ($w=1924$).
 $C_q = 904$ bits $C_l = 813$ bits $\rho = 5.63$.



c) Image approximée ($\alpha=2, T_1=30^2$) et Partition de l'image ($w=1714$).
 $C_q = 900$ bits $C_l = 822$ bits $\rho = 6.31$.

Figure 3.21: Division de l'image du caméraman pour trois valeurs différentes de seuil sur l'erreur quadratique moyenne et une fonction polynomiale de degré 2.

Dans ce qui suit, on a décidé de fixer systématiquement T_1 à 15^2 , indépendamment de l'image traitée.

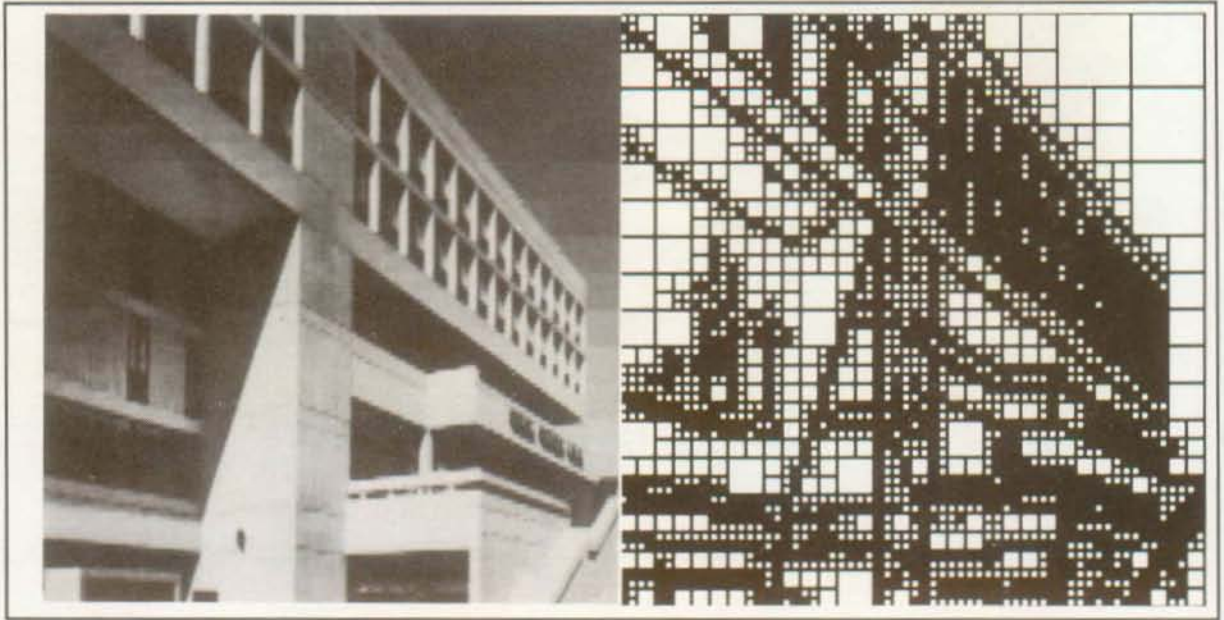
Rôle du degré du polynôme:

Si l'on n'utilise pas d'image de contrôle des contours, l'augmentation du degré du polynôme permet de suivre dans une certaine mesure les sauts de luminance (voir les images des figures 3.13c et 3.14b).

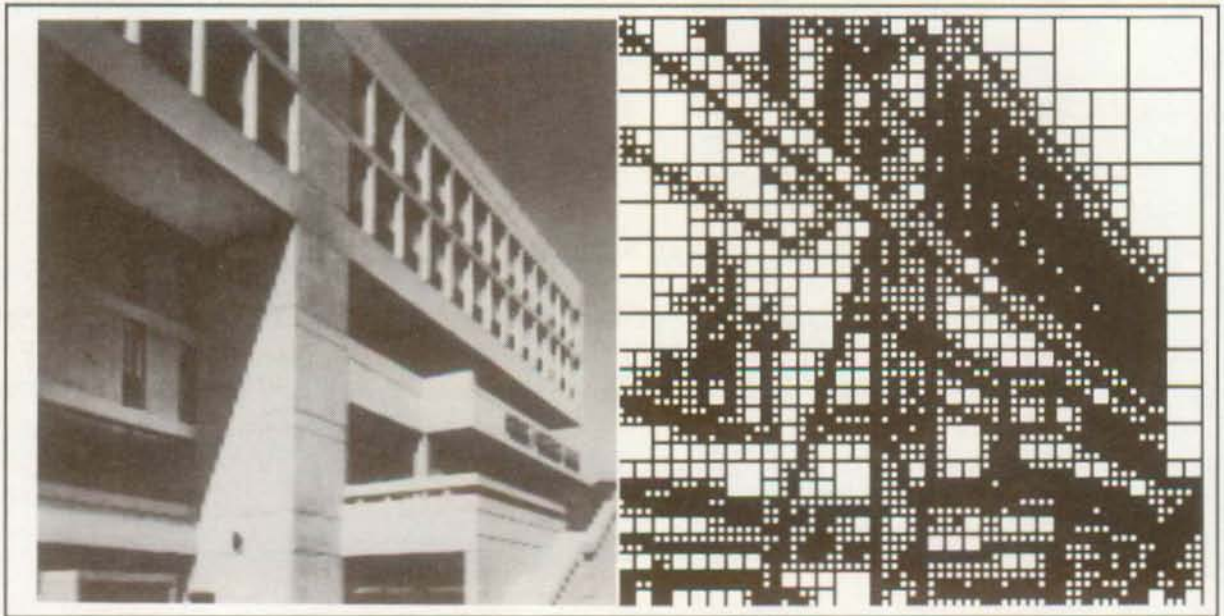
Comme l'algorithme utilisé est celui de la figure 3.7, il paraît plus judicieux de baser le choix du degré du polynôme en liaison avec le rapport de compression. En gardant le même seuil pour T_1 qui permet d'estimer localement la qualité d'un carré ne contenant pas de contour, on choisit le degré du polynôme qui permet d'aboutir à une partition de l'image correspondant à un coût minimal en nombre de bits. Ce coût est toutefois difficile à estimer sans quantifier d'une façon vectorielle les coefficients d'approximation.

Comme ce sujet n'a pas été abordé au niveau de l'algorithme de division, on peut remarquer que le degré du polynôme d'approximation doit être choisi de façon à reproduire la plupart des caractéristiques de la scène. Le problème reste que l'algorithme ne peut savoir à l'avance quelles sont les caractéristiques de luminance de chaque région. Plus on augmente le degré du polynôme, plus le nombre de surfaces que l'on peut décrire augmente. Ceci se fait cependant au détriment de la compression puisque chaque carré sera caractérisé par un nombre plus élevé de coefficients non absolument nécessaires. En utilisant successivement des polynômes séparables de degrés respectifs 0 et 1, d'une part, 1 et 0, d'autre

1 et 0, d'autre part, la figure 3.22 indique les défauts introduits au niveau de l'image du bâtiment. Cette figure est à comparer à la figure 3.20. L'augmentation du degré selon x ou y permet de garder dans une certaine mesure la continuité de la luminance entre les différents carrés de grande taille.



a) Image approximée ($r_1=1, r_2=2, T_1=15^2$) et Partition de l'image ($w=7111$).
 $C_q = 3640$ bits $C_l = 3311$ bits $\rho = 4.48$.



b) Image approximée ($r_1=2, r_2=1, T_1=15^2$) et Partition de l'image ($w=7087$).
 $C_q = 3636$ bits $C_l = 3304$ bits $\rho = 4.49$.

Figure 3.22: Division de l'image du bâtiments pour des fonctions polynomiales séparables de degré 1 selon un axe et 0 selon l'autre.

En observant ces figures, on remarque que la qualité n'est pas parfaite au niveau des carrés de plus petite taille (2x2) contenant des contours car le nombre de coefficients (2) est inférieur au nombre de points contenu dans chacun de ces carrés.

Aspects liés à la compression de l'information:

Ce paragraphe discute de la compression qui est réalisée à ce niveau de la segmentation. En ce qui concerne le codage de position et de taille des carrés, on remarque que le codage conditionnel par taille de carrés est souvent plus efficace que le codage par arbre quaternaire, d'environ 10%. Ceci confirme les hypothèses faites au paragraphe 3.4.1.

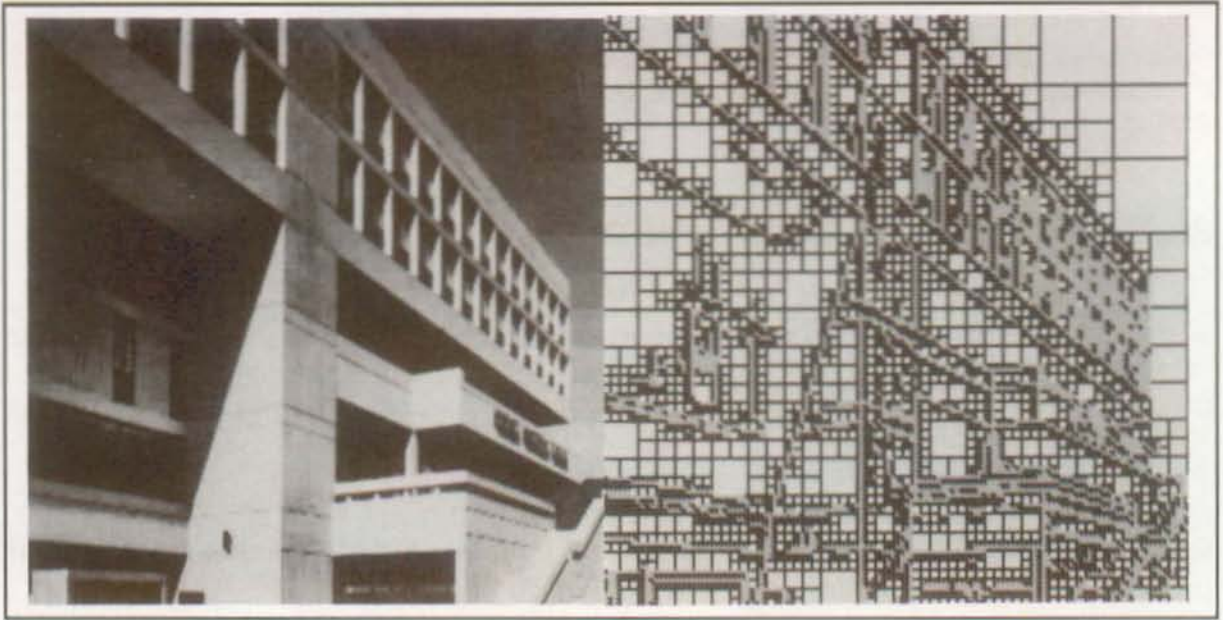
La méthode de codage des coefficients polynomiaux n'est pas optimale ainsi qu'il a été souligné à la section 3.4. Les images présentées aux paragraphes précédents indiquent à chaque fois le rapport de compression ρ .

Comme dans le codage des coefficients polynomiaux, on ne tient absolument pas compte de la corrélation qui existe entre les différents carrés, on présente dans ce qui suit les résultats obtenus par le CDC en approximant chaque carré par sa valeur moyenne (polynôme de degré 0). L'espace des différences est quantifié à 16 niveaux. Après codage de Huffman de ces 16 symboles, le rapport de compression s'exprime par la relation:

$$\rho = 256^2 \times 8 / (C_c + \min(C_q, C_l)) = 524288 / (C_c + \min(C_q, C_l)) \quad (3.20)$$

où C_c représente le coût lié à la transmission des valeurs moyennes de chaque carré.

Les figures 3.23 à 3.25 montrent la reconstitution que l'on obtient en adoptant cette stratégie pour les images du bâtiment, du caméraman et du couple. Chaque figure présente successivement la partition de l'image en carrés (a), sa reconstruction sans quantification de l'espace des différences (b) et l'image codée et restaurée (c).



a) Image approximée ($\alpha=0, T_1=15^2$)

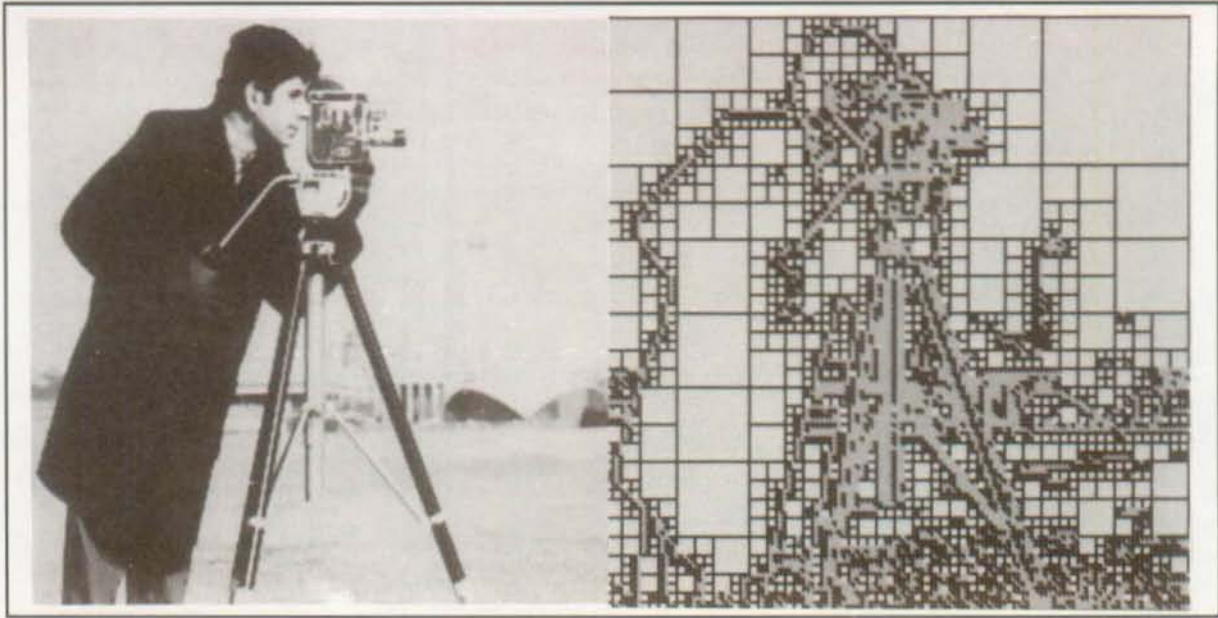
b) Partition de l'image ($w=17635$).

$C_q = 10016$ bits $C_l = 9933$ bits $\rho = 12.62$.



c) Image codée et rehaussée.

Figure 3.23: Division de l'image du bâtiment avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits.



a) Image approximée ($\alpha=0, T_1=15^2$)

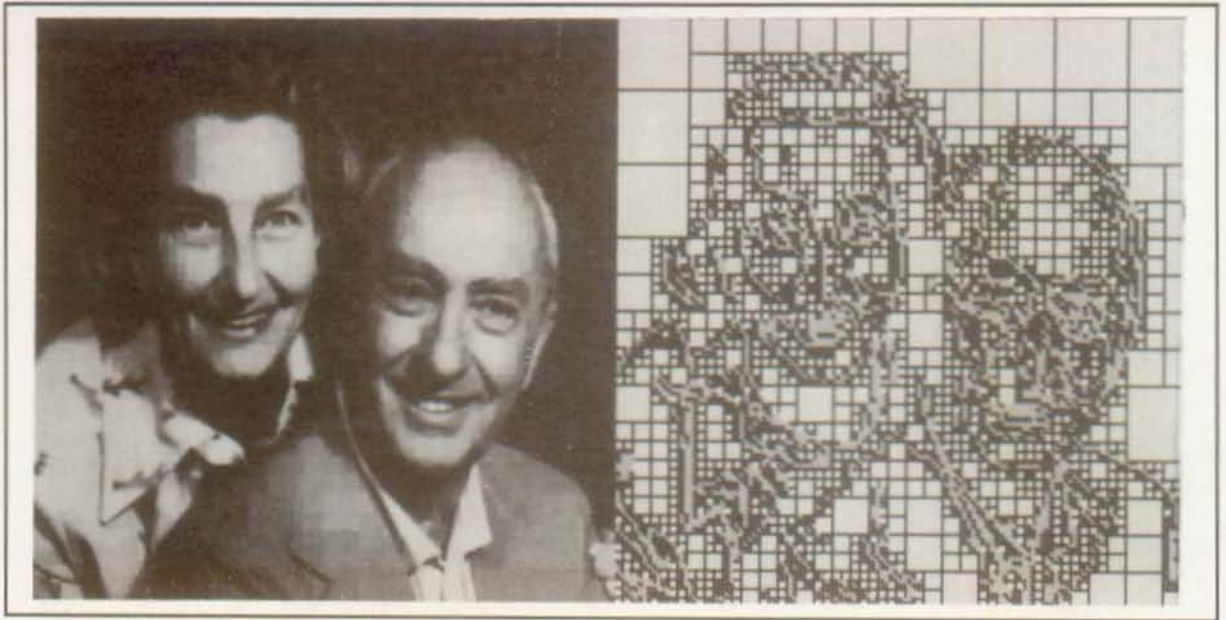
b) Partition de l'image ($w=14239$).

$C_q = 7844$ bits $C_l = 7765$ bits $\rho = 15.58$.



c) Image codée et rehaussée.

Figure 3.24: Division de l'image du cameraman avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits.

a) Image approximée ($\alpha=0, T_1=15^2$)b) Partition de l'image ($w=13903$).

$C_q = 8756$ bits $C_l = 8952$ bits $\rho = 17.96$.



c) Image codée et rehaussée.

Figure 3.25: Division de l'image du couple avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits.

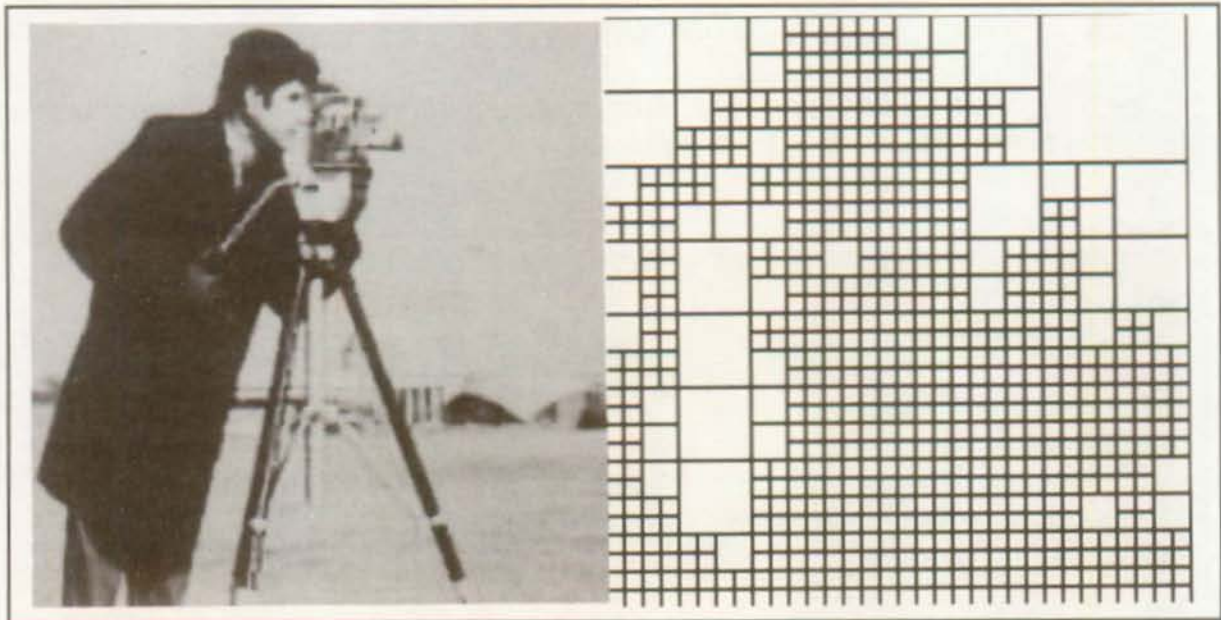
Il est important de remarquer que le codage de position permettant de reconstituer la structure de la segmentation après division représente une part négligeable du coût associé à la reconstitution de l'image approximée. Certes, la quantification des coefficients n'a pas été étudiée dans les détails pour réduire au mieux la corrélation qui existe entre l'ensemble de coefficients associé à chaque carré. Il n'en reste pas moins que la structure simple de la segmentation après division rend son codage peu coûteux. Après le processus de rassemblement, la

situation sera inversée. Le codage des coefficients prendra une part négligeable par rapport au codage de forme et de position de chaque région.

3.5.3 Limitation de la plus petite taille de carré:

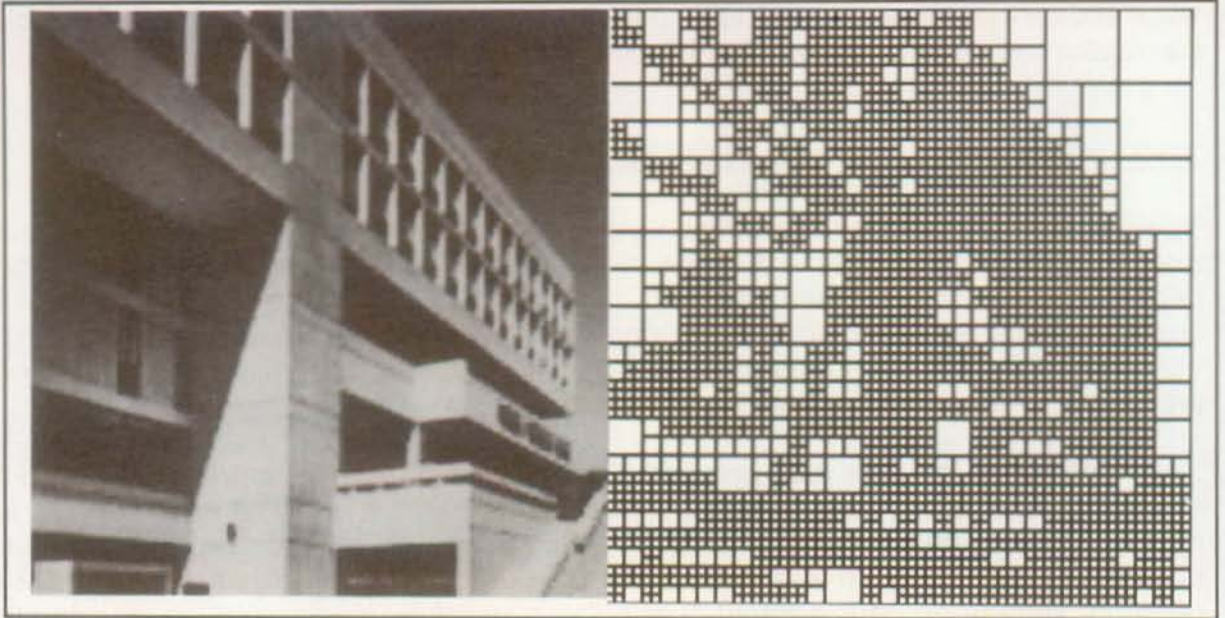
Lorsque le nombre de coefficients du polynôme n'est pas beaucoup plus petit que le nombre de points à l'intérieur d'un carré, on a vu que l'approximation pouvait suivre dans une certaine mesure le signal original même lorsque le carré considéré contient un contour. Si on décide de limiter la plus petite taille de carré, on risque d'avoir une représentation relativement fidèle du signal original tout en diminuant sensiblement le nombre de carrés à coder.

A cet effet, la figure 3.26 présente l'image du caméraman approximée par des polynômes de degré 3 où la plus petite taille de carré a été limitée à 8×8 . De la même manière, on présente à la figure 3.28 l'image du bâtiment approximée par des plans lorsqu'on a limité la plus petite taille de carré à 4×4 . Enfin, la figure 3.28 présente l'image du couple sachant que l'on a utilisé le même schéma de codage que pour la figure 3.25 alors que les carrés les plus petits sont de taille 2×2 .



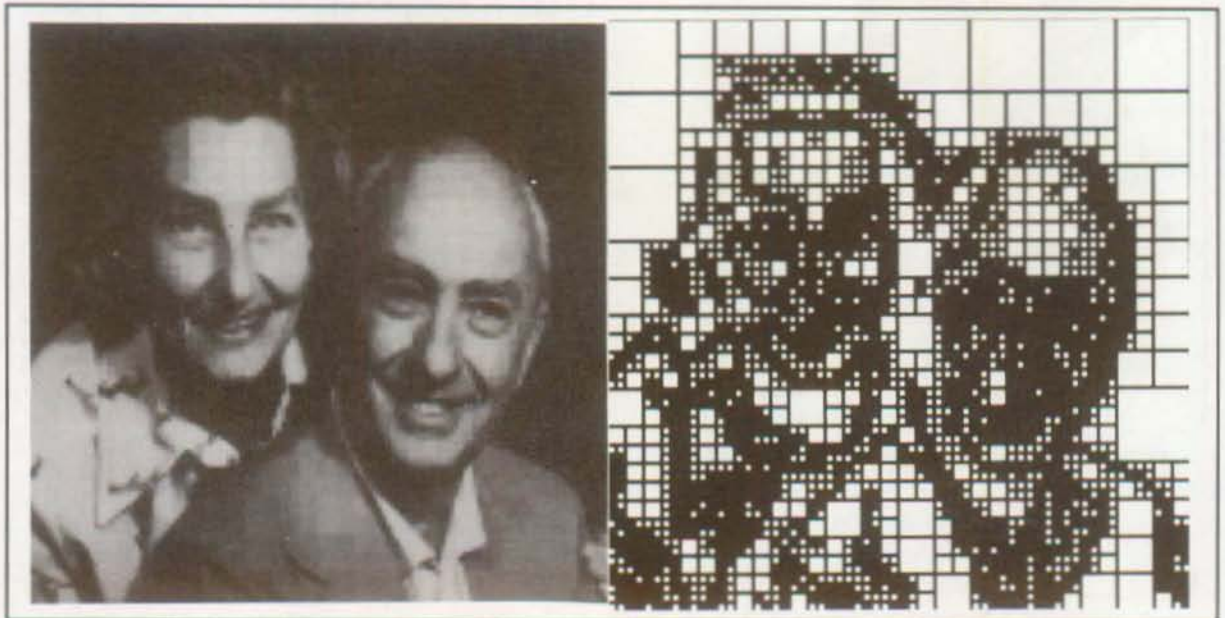
a) Image approximée ($\alpha=3$ $T_1=15^2$) b) Partition de l'image ($w=682$).
 $C_q = 272$ bits $C_l = 232$ bits $\rho = 9.57$.

Figure 3.26: Division de l'image du caméraman avec une fonction d'approximation par polynômes de degré 3. La division est limitée à des carrés de taille 8×8 .



a) Image approximée ($\alpha=3$ $T_1=15^2$) b) Partition de l'image ($w=2713$).
 $C_q = 1140$ bits $C_l = 954$ bits $\rho = 7.94$.

Figure 3.27: Division de l'image du bâtiment avec une fonction d'approximation par polynômes de degré 1. La division est limitée à des carrés de taille 4x4.



a) Image approximée ($\alpha=0$, $T_1=15^2$) b) Partition de l'image ($w=6571$).
 $C_q = 3256$ bits $C_l = 3001$ bits $\rho = 41.02$.



c) Image codée et rehaussée.

Figure 3.28: Division de l'image du couple avec une fonction d'approximation par niveaux constants. Image codée et rehaussée par CDC 16 bits. Plus petite taille de carrés = 2×2 .

3.6 Conclusion :

Dans ce chapitre, on a étudié la première phase de la segmentation : la division de l'image en un ensemble de carrés de différentes tailles, chacun d'eux correspondant à une partie d'espace régulier.

Dans un premier temps, le critère de division a été défini en termes généraux: un carré est divisé chaque fois que l'approximation polynomiale des données qu'il contient ne permet pas de reproduire l'information contour (l'erreur de contour est trop élevée), ou lorsque l'erreur quadratique moyenne dépasse un certain seuil.

Dans un deuxième temps, on a présenté des implantations possibles de l'algorithme suivies d'une analyse de la complexité des traitements à effectuer. De celle-ci, il ressort qu'il est possible de simplifier considérablement le problème en remplaçant la mesure sur l'erreur de contour par un simple test de présence d'un contour à l'intérieur d'un carré donné.

La section suivante traite de la représentation de l'information segmentée après application de l'algorithme de division. Deux codes sont définis pour décrire la structure de la segmentation: le premier utilise la notion d'arbre quaternaire; le second utilise la probabilité d'apparition de chaque taille de carré tout en conservant les contraintes géométriques imposées par l'algorithme de division. Lorsque certaines hypothèses sont valables, ce qui est souvent le cas (voir section 3.5), ce code s'avère plus efficace (10%) que le précédent. Le codage de l'approximation n'est pas traité en détail dans ce chapitre ce qui réduit les rapports de compression que l'on peut estimer après division. Un code simple a été cependant développé pour diminuer la redondance existant entre les

approximations des différents carrés dans le cas d'une approximation par constantes.

Enfin, la dernière section illustre expérimentalement les différents paramètres de l'algorithme. En particulier, la forme de la segmentation apparaît plus régie par les conditions imposées par l'image de contrôle des contours que par le seuil placé sur la mesure de l'erreur quadratique moyenne. L'augmentation du degré du polynôme ne fait que limiter la plus petite taille de carré sans changer la partition de l'image pour les régions de plus grande taille. La texture correspondant aux carrés de grande taille est cependant plus fidèlement reproduite en augmentant le degré du polynôme; pour les carrés de plus petite taille contenant des contours, on peut voir apparaître certains effets de blocs lorsque le nombre de coefficients d'approximation reste trop inférieur au nombre de points du carrés. Cette section donne aussi à titre indicatif les rapports de compression qui peuvent être obtenus à ce niveau. En utilisant une approximation par constantes, une reconstitution acceptable de l'image du couple a pu être obtenue avec un rapport de compression dépassant 40 à 1. Il nous reste à voir comment ceci peut être amélioré en réduisant encore la redondance d'information par le processus de rassemblement.

4. Opération de rassemblement

Dans ce chapitre, on aborde la deuxième phase de la segmentation : *le rassemblement* ou *le regroupement*. A partir de la segmentation obtenue par l'opération de division, les différents carrés vont être rassemblés de manière à définir les espaces réguliers de l'image. La segmentation finale résulte en un ensemble de régions dont les frontières correspondent au mieux aux contours réels des objets contenus dans l'image.

A chaque étape du processus de rassemblement, on associe les deux régions présentant la plus faible mesure de dissimilarité. Pour tenir compte de la structure des régions impliquées dans le rassemblement, il faut faire évoluer cette mesure. Les régions placées dans le voisinage de contours réels seront traitées de façon particulière de manière à garantir la précision requise pour décrire ces contours dans l'image approximée.

Six sections forment le contenu de ce chapitre. La première définit le principe de l'algorithme, les critères de dissimilarité utilisés et la condition d'arrêt du processus. La deuxième section décrit comment il est possible d'obtenir l'approximation optimale au sens des moindres carrés d'un signal sur un domaine à partir des approximations selon ce même critère calculées sur un ensemble de domaines définissant une partition quelconque du domaine de départ. Elle montre comment cette propriété s'applique efficacement à l'opération de rassemblement. Cette section traite également de l'estimation des critères de dissimilarité. La troisième section étudie la structure des données qui peut être utilisée pour implanter efficacement l'algorithme de rassemblement. La section suivante aborde le problème de la complexité de cet algorithme et propose des solutions sous-optimales pour simplifier certaines phases du rassemblement. La cinquième section définit la stratégie de rassemblement dans son ensemble; elle explique notamment comment faire évoluer les mesures de dissimilarité avec la structure des régions impliquées dans le rassemblement et illustre sur un exemple les différentes phases de la procédure à suivre. Enfin, le chapitre se termine en rappelant les points essentiels de l'exposé qui précède.

4.1 Principe de l'algorithme :

Après le processus de division, l'image est constituée par un ensemble de carrés définissant pour la plupart (en tout cas ceux qui ne contiennent pas de contours) des parties d'espaces réguliers de l'image. Il s'agit de les rassembler maintenant de manière à faire apparaître les objets de la scène. Le critère de regroupement est basé sur la notion de graphe de contiguïté de région (GCR) décrite au chapitre 2.

Ainsi, deux régions voisines de l'image initialement segmentée par l'algorithme de division sont rassemblées si la mesure de dissimilarité associée à la branche du GCR qui les relie est la plus faible par rapport à l'ensemble des autres mesures de dissimilarité du graphe. Ce principe est itéré autant de fois que nécessaire jusqu'à ce que tout nouveau rassemblement entraîne une distorsion inacceptable de l'image originale, l'approximation sur la nouvelle région n'étant plus en mesure de reproduire avec une qualité suffisante le signal original (par exemple, perte de l'information contour). Appelons N_r le nombre de régions dont

est formée la segmentation finale. Parmi toutes les partitions possibles de l'image en N_r régions, chacune d'elle étant approximée par un polynôme bidimensionnel, celle qu'on vient d'obtenir par regroupement systématique des deux régions les plus ressemblantes est optimum pour le critère de dissimilarité choisi.

On examine dans ce qui suit comment il est possible de satisfaire aux exigences de qualité du chapitre 2. Il reste également à définir sur quelle base sont estimés les critères de dissimilarité existant entre deux régions voisines et comment peut-on décider qu'un nouveau rassemblement entraînerait une distorsion inacceptable de l'image originale.

4.1.1 Précision dans la description des frontières de région:

Le processus de rassemblement décrit jusqu'ici est relativement simple. Néanmoins, en observant les partitions obtenues par l'algorithme de division, on remarque que l'image n'est pas subdivisée au-dessous d'une certaine taille, cette taille étant définie en liaison avec l'existence d'une approximation optimale au sens des moindres carrés (voir inéquation (3.1)). Comment garantir alors la précision souhaitée au niveau des frontières des régions de la segmentation finale? En appliquant l'algorithme de rassemblement tel quel, on serait limité par la plus petite taille de carré imposée par l'algorithme de division. Ceci est illustré à la figure 4.1.

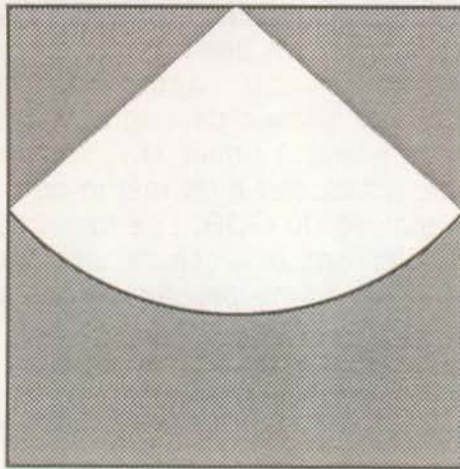
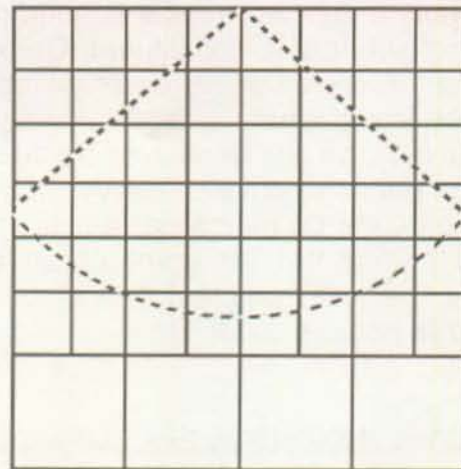
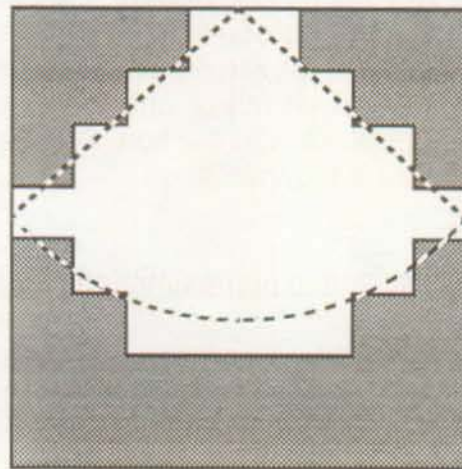


Image originale
(4 espaces réguliers)



Partition après division
 □ Plus petite taille de carré



Segmentation finale

$$(N_r = 4)$$

Figure 4.1: Erreur de frontière commise lors de l'application de l'algorithme de rassemblement à la partition initiale donnée par l'algorithme de division.

On remarque que dans la partition initiale, il existe toute une série de carrés qui ne correspondent pas à des parties d'espaces réguliers de l'image originale. Ce sont les carrés contenant des contours. Ils sont constitués de points appartenant à deux objets différents, voire plus, de l'image. Pour satisfaire aux exigences de correspondance entre frontières de régions et contours réels des objets, ces points doivent être associés à ces objets. Définir des régions à part entière pour des groupes de ces points entraîne l'apparition de faux contours inutiles dont le codage est coûteux. Chaque point est inséré au domaine dont les caractéristiques lui correspondent le plus.

Pour ce faire, on définit dans un premier temps quels sont les carrés de la partition initiale contenant des contours en utilisant l'image de contrôle

correspondante. On modifie ensuite le GCR en remplaçant chacun de ces carrés par les points qui le constituent. Ceux-ci définissent dès lors des noeuds du GCR. Comme l'association de deux de ces points est impossible, l'approximation ne pouvant être définie de façon univoque (pour des polynômes de degré supérieur ou égal à 1), on fixe la mesure de dissimilarité entre points à l'infini. Une fois inséré dans un domaine qui leur est voisin, il est possible d'associer à ce même domaine d'autres points du même carré initial par simple analyse du GCR. Les mesures de dissimilarité entre un point et un domaine avoisinant définissant une partie d'espace régulier peuvent être établies en utilisant la même procédure que pour les autres noeuds du GCR.

4.1.2 Choix des critères de dissimilarité entre régions:

Les mesures de dissimilarité doivent permettre d'estimer la correspondance qui existe entre deux régions voisines de l'image avant rassemblement. Comme il a été décrit à la section 2.1, elles doivent se faire en liaison avec l'approximation choisie pour représenter les données à l'intérieur de chaque région. Elles ne peuvent cependant pas se faire indépendamment de la structure des objets ou des parties d'objet auxquels ces régions correspondent. Comme ceux-ci ne sont pas à priori connus avant de commencer le processus de rassemblement, on est amené à définir toute une série de mesures. En liaison avec la forme et la taille des régions impliquées dans un rassemblement on utilise une mesure plutôt qu'une autre. Ce problème sera étudié dans les détails à la section 4.5. On élabore ici une liste de ces différentes mesures et de leurs propriétés.

Mesure de distance entre coefficients d'approximation u_i sur chaque région:

Une première possibilité consiste à comparer les coefficients d'approximation relatifs aux deux régions contiguës. Pour ce faire, on estime une certaine distance qui existe entre les deux vecteurs correspondants que l'on note u_1 et u_2 .

Cependant, cette mesure de distance ne tient absolument pas compte des valeurs prises par les fonctions d'approximation sur chaque région. Celles-ci sont liées aux valeurs de u_1 et u_2 mais aussi à la forme des régions. Or, ce sont ces valeurs qui sont représentatives de la correspondance qui existe entre les deux régions.

En utilisant une telle mesure, on ne prend en outre pas en considération la modification de la fonction d'approximation une fois que les deux régions auront été regroupées. Examinons d'autres mesures qui l'utilisent et qui s'avèrent donc plus performantes.

Erreur quadratique, respectivement erreur quadratique moyenne par rapport au signal original:

Cette mesure de dissimilarité notée EQ, respectivement EQM permet d'estimer la dégradation qui est introduite par le rassemblement de deux régions au niveau de l'approximation selon le critère des moindres carrés du signal original.

Lorsqu'on pondère l'erreur quadratique par le nombre de points de la région après rassemblement, on se rend indépendant de la taille de celle-ci. Selon la taille moyenne des régions du GCR, cette possibilité peut s'avérer utile pour empêcher le rassemblement de deux petites régions très différentes, EQM étant plus faible sur ces deux régions que pour deux grandes régions relativement semblables à cause de la pondération par le nombre de points. EQM tend également à favoriser l'insertion de très petites régions dans des régions de taille beaucoup plus grande indépendamment du signal qui caractérise cette petite région. En effet, si l'on considère le regroupement potentiel d'une grande région avec une petite région qui lui est très différente, les coefficients de la nouvelle approximation au sens des moindres carrés sur l'ensemble de ces deux régions sont très proches de ceux de l'approximation sur la région de grande taille. Les erreurs introduites au niveau de la petite région apparaissent comme négligeables à cause de la pondération par le nombre total de points. Il y a dès lors de fortes chances que le rassemblement s'effectue entre ces deux régions avant de s'effectuer sur deux régions qui diffèrent moins l'une de l'autre et dont la taille après rassemblement est plus petite.

Le choix de EQ permet d'éviter de considérer des régions de très grande taille au même titre que de petites régions, correspondant souvent à des détails. Cependant, prendre une telle mesure empêche à la longue d'obtenir des régions dépassant une certaine taille, car même si l'erreur d'approximation reste très faible sur beaucoup de points, les multiples contributions de tous ces points tendent à donner à EQ une valeur importante. Ce désavantage peut être supprimé en considérant l'erreur quadratique commise entre le signal approximé avant rassemblement et après rassemblement, les contributions de l'erreur en chaque point étant significatives.

Erreur quadratique, respectivement erreur quadratique moyenne entre signal approximé avant regroupement et signal approximé après regroupement:

On note cette dissimilarité EQC, respectivement EQMC où C signifie "courante". EQMC s'obtient de EQC par la division de cette dernière par le nombre de points de la région après rassemblement. Ce genre de critère permet d'estimer la modification exacte introduite à chaque étape de rassemblement par ce dernier.

Supposons avoir une certaine segmentation de l'image où chaque région est approximée par une certaine fonction bidimensionnelle. Si l'on peut dire que cette segmentation est optimale en ce qu'elle n'introduit aucune distorsion appréciable de l'image originale, il est raisonnable d'admettre de mesurer la dégradation de l'image par rapport à la modification de cette segmentation optimale, par exemple par rassemblement de deux régions. Si la distorsion n'est pas appréciable, cela ne veut pas pour autant dire que l'erreur quadratique entre l'approximation et le signal original soit nulle. Cependant le critère d'optimalité de la segmentation permet de la considérer comme telle. Ceci peut être par exemple le cas lorsque l'approximation fait disparaître la granularité locale d'une certaine région. L'approximation sert de modèle pour décrire la partie d'espace régulier délimitée par chaque région de la segmentation. Tout se passe comme si l'image originale est remplacée avant chaque étape de rassemblement par l'image approximée. Une telle mesure peut seulement être utilisée avec succès tant que le processus de rassemblement garantit la qualité de l'image au sens de la section 2.1.

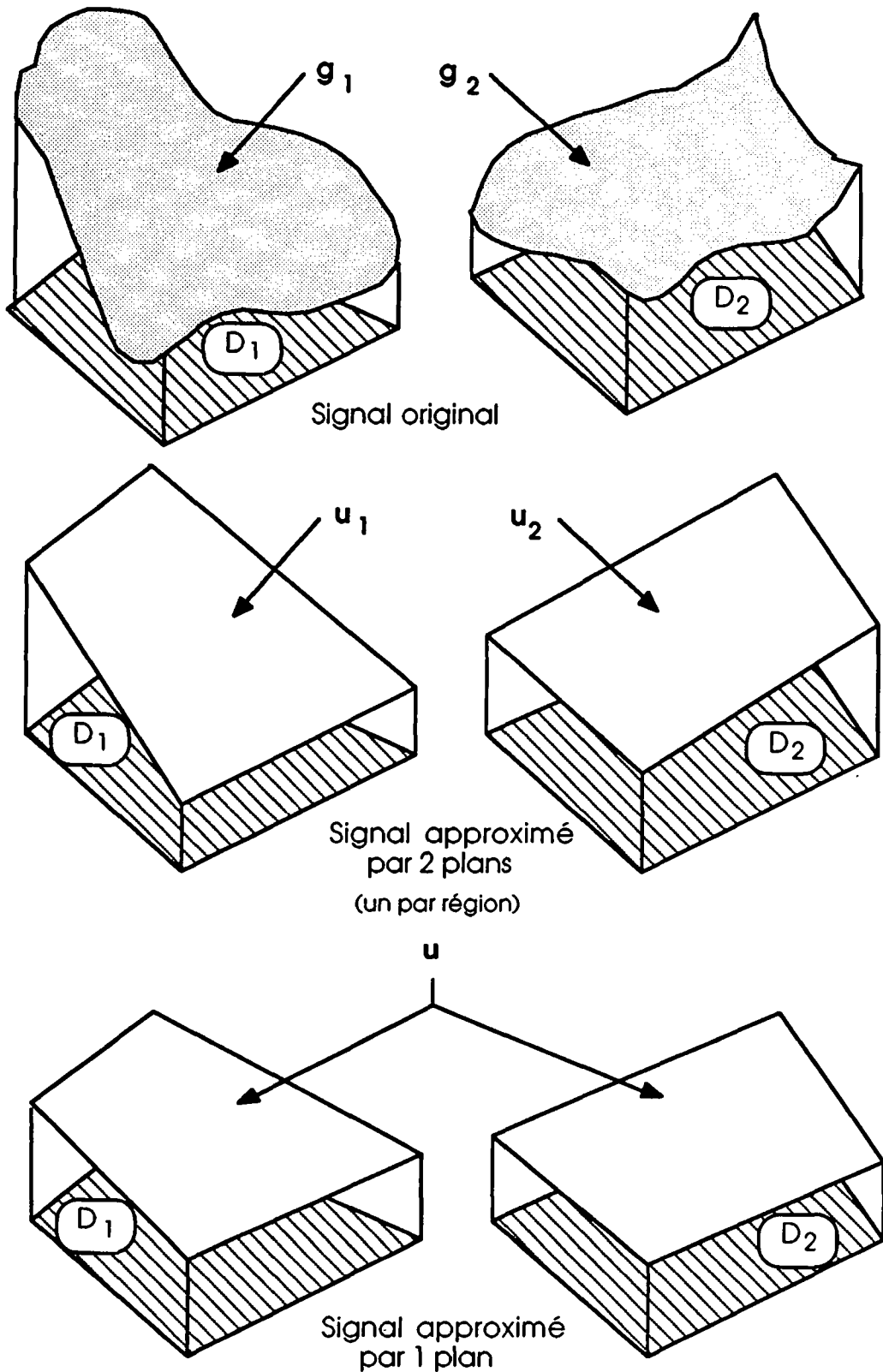


Figure 4.2: Propriété de décomposition de l'approximation au sens des moindres carrés.

Etant donné un signal défini sur deux domaines D_1 et D_2 , l'approximation optimale au sens des moindres carrés du signal sur l'ensemble de ces deux domaines est identique à celle que l'on obtiendrait en la calculant par rapport aux approximations

au sens des moindres carrés calculées sur chaque domaine pris séparément. Dans l'exemple ci-dessus, le résultat correspondant au vecteur des coefficients d'approximation u peut être obtenu indifféremment des approximations représentées par les vecteurs u_1 et u_2 , ou des signaux originaux correspondant à g_1 et g_2 (la figure indique une approximation par plan).

Ce choix est renforcé par le fait que l'approximation au sens des moindres carrés sur un ensemble de régions d'un certain signal est identique à l'approximation au sens des moindres carrés calculée par rapport aux approximations optimales de ce même signal sur chaque région. Ceci est illustré par la figure 4.2. Ainsi, EQC correspond à l'accroissement d'erreur quadratique qui existe avant et après rassemblement entre deux régions. Elle représente donc parfaitement la modification de signal introduite par le rassemblement de deux régions. Cette propriété sera démontrée à la section suivante où l'on discute du calcul des approximations et des critères d'erreur.

Parmi les différents critères proposés jusqu'ici, il en est aucun qui définit une mesure qui soit plus ou moins indépendante de la dimension des deux régions impliquées par un rassemblement. On a ainsi été amené à recourir à la mesure de dissimilarité suivante.

Erreur quadratique pondérée, respectivement erreur quadratique courante pondérée par le nombre de points de chaque région:

Il s'agit simplement de diviser les contributions à l'erreur quadratique sur chaque domaine par le nombre respectif de points de chaque domaine. Bien entendu, les erreurs quadratiques sont estimées pour l'approximation optimale au sens des moindres carrés sur l'ensemble des deux domaines.

EQCP correspond à l'erreur quadratique courante pondérée. Elle se distingue de EQP (erreur quadratique pondérée) par le simple fait qu'elle est calculée entre les signaux approximés avant et après rassemblement.

Ce critère ne s'avère pas être optimum dans tous les cas. En effet, une toute petite région correspondant à un détail insignifiant de l'image n'est pas insérée dans une région de bien plus grande taille par le simple fait que l'erreur d'approximation apparaît comme très grande sur ce petit domaine.

Remarque:

De tout ce qui précède aucun critère de dissimilarité lié à l'information contour n'a été proposé. Il apparaît en effet difficile d'utiliser une telle mesure pour deux raisons. D'une part, les calculs nécessaires à son estimation sont prohibitifs car il est indispensable de connaître la valeur exacte du signal d'approximation au voisinage du contour traité, ce qui n'est pas le cas pour les autres critères. D'autre part, l'importance d'un contour est très dépendante du contexte. L'existence d'un contour entre deux régions ne veut pas dire que ces deux régions ne peuvent jamais être rassemblées, surtout si l'on désire atteindre des rapports de compression élevés. Tout dépend de l'importance relative de ce contour. A la

section 4.5, on verra que la qualité des résultats sera garantie en insérant les régions qui possèdent des contours à un moment adéquat dans le GCR.

4.1.3 Critère d'arrêt du processus:

Juger à partir de quand un rassemblement entraîne une distorsion inacceptable de l'image originale est une tâche difficile. En effet, tout dépend de l'image traitée. On peut cependant dire que l'on cherche à atteindre une segmentation comprenant un nombre de régions aussi réduit que possible. Ceci, dans le but d'avoir une représentation aussi compacte que possible de la scène originale, soit pour des applications d'analyse ou d'interprétation de scène, soit pour du codage.

On peut décider de limiter tout simplement le nombre de régions à partir duquel on a atteint une segmentation jugée relativement simple. En adoptant une telle démarche, on ne garantit en rien de préserver le contenu sémantique de l'image originale. Par contre, le rapport de compression atteint sera d'autant plus élevé que le nombre de régions de la segmentation est faible. Cette solution sera donc choisie.

Commentaire:

Fixer un seuil sur une certaine mesure d'erreur telle qu'erreur quadratique totale sur l'image, dissimilarité entre deux régions ne s'est jamais révélé efficace. En effet, pour une même valeur de seuil sur cette mesure d'erreur, le nombre de régions de la segmentation finale (et donc le rapport de compression) varie considérablement d'une image à une autre.

Après avoir défini les critères de dissimilarité et d'arrêt du processus de rassemblement, il s'agit d'étudier mathématiquement l'évolution de l'approximation au sens des moindres carrés après rassemblement de deux régions.

4.2 Evolution de l'approximation après rassemblement :

Cette section démontre la *propriété de décomposition de l'approximation* optimale au sens des moindres carrés. Celle-ci stipule qu'il est possible d'obtenir l'approximation selon ce critère sur un certain domaine D à partir des approximations calculées selon ce même critère sur un ensemble de domaines qui définissent une partition de D .

Il est ensuite montré comment une telle propriété peut se révéler très efficace dans le cadre de l'algorithme de rassemblement. Enfin, le dernier paragraphe illustre comment il est possible d'estimer les différents critères de dissimilarité par l'utilisation d'une telle propriété.

4.2.1 Décomposition de l'approximation au sens des moindres carrés:

Un exemple illustrant cette propriété a été présenté à la figure 4.2. Pour prouver cette propriété, il suffit de montrer qu'elle est vraie en divisant un domaine

D en deux domaines disjoints D_1 et D_2 . En adoptant une notation similaire à celle définie au chapitre 2, appelons u , u_1 et u_2 les vecteurs des coefficients d'approximation pour les domaines D , D_1 et D_2 , respectivement sachant que l'on utilise le même ensemble $\Psi_i(x,y)$ de fonctions de base. De la même façon, on définit par N , N_1 et N_2 le nombre de points de chaque domaine. Bien entendu, on a la relation $N=N_1+N_2$. Le vecteur des pixels g correspondant au domaine D s'obtient simplement par la concaténation des vecteurs de pixels g_1 et g_2 relatifs aux domaines D_1 et D_2 . On définit de façon similaire pour chaque domaine les matrices Z de taille $N \times r$, Z_1 de taille $N_1 \times r$ et Z_2 de taille $N_2 \times r$; les matrices carrées de taille $r \times r$ S , S_1 et S_2 ainsi que les vecteurs h , h_1 et h_2 .

En appliquant la procédure de minimisation des moindres carrés, on obtient les équations suivantes:

$$(Z^T Z).u = Z^T g \Leftrightarrow S u = h \quad (4.1)$$

$$(Z_1^T Z_1).u_1 = Z_1^T g_1 \Leftrightarrow S_1 u_1 = h_1 \quad (4.2)$$

$$(Z_2^T Z_2).u_2 = Z_2^T g_2 \Leftrightarrow S_2 u_2 = h_2 \quad (4.3)$$

La relation (4.1) a été obtenue par minimisation de l'erreur quadratique EQ sur le domaine D donnée par:

$$EQ = (g - g^\wedge)^T (g - g^\wedge) = (g_1 - g_1^\wedge)^T (g_1 - g_1^\wedge) + (g_2 - g_2^\wedge)^T (g_2 - g_2^\wedge) \quad (4.4)$$

où g^\wedge , g_1^\wedge et g_2^\wedge représente les vecteurs des pixels approximatés sur chacun des domaines lorsque l'approximation est calculée sur l'ensemble du domaine D . En faisant apparaître le vecteur des coefficients u , l'équation (4.4) peut s'écrire en supposant que l'on ait choisi une origine unique de manière à ce que les matrices Z_1 et Z_2 soient définies pour cette origine:

$$EQ = (g_1 - Z_1 u)^T (g_1 - Z_1 u) + (g_2 - Z_2 u)^T (g_2 - Z_2 u) \quad (4.5)$$

On obtient la solution qui minimise les moindres carrés en dérivant EQ par rapport à u , et en annulant l'expression correspondante. On a alors:

$$-2 Z_1^T (g_1 - Z_1 u) - 2 Z_2^T (g_2 - Z_2 u) = 0$$

$$(Z_1^T Z_1 + Z_2^T Z_2) u = Z_1^T g_1 + Z_2^T g_2 \quad (4.6)$$

Si l'on reprend les équations (4.1) à (4.3), on remarque que chaque terme de l'équation (4.6) est connu. La solution optimale s'exprime tout simplement par :

$$u = (S_1 + S_2)^{-1} (S_1 u_1 + S_2 u_2) \quad (4.7)$$

On a ainsi démontré qu'en utilisant l'approximation selon les moindres carrés, il est possible d'obtenir les coefficients d'approximation optimaux sur un domaine D directement de ceux obtenus selon le même critère sur une partition de ce domaine.

Commentaires:

1) Si l'on considère une partition de D formée de n domaines D_1, D_2 à D_n , la relation (4.7) devient en gardant le même genre de notations:

$$u = (S_1 + S_2 + \dots + S_n)^{-1} (S_1 u_1 + S_2 u_2 + \dots + S_n u_n) \quad (4.8)$$

2) La mémorisation des vecteurs h_i et des matrices S_i relatifs à un ensemble de n domaines disjoints permet d'estimer rapidement l'approximation optimale sur la réunion de ces domaines. (On peut bien entendu mémoriser uniquement les vecteurs d'approximation u_i et les matrices S_i). Les paramètres caractéristiques de ce domaine obtenu par réunion peuvent être définis par les équations suivantes:

$$S = \sum_{i=1}^n S_i \quad (4.9)$$

$$h = \sum_{i=1}^n h_i \quad (4.10)$$

$$u = S^{-1} h \quad (4.11)$$

Du point de vue de la complexité de calcul, on peut estimer le vecteur h en $(n-1)r$ additions et la matrice S en $(n-1)r^2$ additions. Si la propriété de décomposition n'était pas valable il aurait fallu $O(Nr^2)$ opérations (voir section 2.3) où N représente le nombre total de points du domaine qui est en général beaucoup plus grand que n , le nombre de domaines. La résolution du système (4.11) reste nécessaire mais elle n'entraîne qu'un nombre réduit d'opérations ($O(r^3)$) par rapport au gain précédemment obtenu.

3) Les matrices S_i caractérisent la forme et la position respective des domaines D_i . Les vecteurs des coefficients u_i caractérisent le signal original.

4) La propriété de décomposition est valable indépendamment du choix des fonctions de base $\Psi_i(x,y)$. Une solution séparable peut être trouvée si le problème de l'approximation est formulable de façon séparable (voir section 3.3). Les paramètres caractéristiques d'un domaine obtenu par réunion de n domaines sont alors donnés par:

$$S_k = \sum_{i=1}^n S_{k(i)} \quad (4.12)$$

$$\mathbf{S}_1 = \sum_{i=1}^n \mathbf{S}_{1(i)} \quad (4.13)$$

$$\mathbf{H} = \sum_{i=1}^n \mathbf{H}_{(i)} \quad (4.14)$$

$$\mathbf{U} = \mathbf{S}_k^{-1} \mathbf{H} \mathbf{S}_1^{-1} \quad (4.15)$$

où l'on a défini les matrices $\mathbf{S}_{k(i)}$, $\mathbf{H}_{(i)}$ et $\mathbf{U}_{(i)}$ de manière semblable au chapitre 3.

5) La propriété de décomposition peut être généralisée à l'analyse de signaux multidimensionnels. Pour un espace à m dimensions, les fonctions de base sont à m variables et les domaines D sont à m dimensions. Pour utiliser le même formalisme, il suffit de placer les échantillons du signal multidimensionnel dans un vecteur \mathbf{g} .

4.2.2 Application à l'algorithme de rassemblement:

Il s'agit maintenant d'étudier comment la propriété de décomposition de l'approximation au sens des moindres carrés peut se révéler utile dans le contexte de l'algorithme de rassemblement.

Si l'on considère le point de départ de cet algorithme, c'est-à-dire la partition de l'image par l'algorithme de division, on remarque que l'approximation au sens des moindres carrés a été calculée par rapport à une origine relative à chaque carré. Lorsqu'on cherche l'approximation de deux régions voisines, on ne peut donc appliquer telles quelles les équations (4.9) à (4.11) sans avoir explicité les paramètres caractéristiques des deux régions (matrices \mathbf{S}_1 et \mathbf{S}_2 ; vecteurs \mathbf{h}_1 , \mathbf{h}_2 , \mathbf{u}_1 et \mathbf{u}_2), par rapport à une origine unique. La première étape consiste donc à exprimer les paramètres caractéristiques de tous les carrés correspondant au GCR initial par rapport à une origine unique. Pour éviter le plus possible les problèmes d'arrondis de calcul liés au choix des fonctions polynômiales, on a pris comme origine le centre de géométrie du carré de l'image.

Une fois ce changement d'origine réalisé, on remarque que la propriété de décomposition est parfaitement applicable pour le calcul de l'approximation optimale après chaque étape de rassemblement. Les seules contraintes existantes sont imposées au niveau des régions qui peuvent être regroupées: celles-ci doivent être contiguës.

La propriété de décomposition peut s'avérer utile pour le choix du lieu de rassemblement. En effet, en décidant de lier approximation et segmentation, il est nécessaire d'estimer l'expression de celle-ci sur toutes les combinaisons possibles de deux régions voisines, et d'associer une mesure de dissimilarité permettant de déterminer la correspondance qui existe entre toutes ces régions, prises deux à

deux. Regardons justement comment il est possible d'estimer les différents critères de dissimilarité en utilisant cette propriété.

4.2.3 Evaluation des critères de dissimilarité:

Les notations adoptées jusqu'ici relatives à deux régions restent parfaitement valables. On définit également par EQ_1 et EQ_2 , les erreurs quadratiques qui existent pour chaque région entre le signal original et son approximation calculée relativement à chaque région.

Vu les considérations effectuées à la section précédente, tous les critères de dissimilarité vont être parcourus mis à part la mesure de distance existant entre coefficients d'approximation de deux régions.

Pour ces estimations, une propriété fondamentale de la section 2.4 est à reprendre ici. L'erreur quadratique minimale s'exprime par la différence entre l'énergie du signal original et celle du signal approximé exprimé par $\mathbf{u}^T \mathbf{h}$. Appelons E l'énergie du signal original, E_1 et E_2 celles des signaux originaux sur chaque région, E^\wedge celle de l'approximation sur l'ensemble des deux régions, E_1^\wedge et E_2^\wedge les énergies des signaux approximés sur chaque région, les approximations ayant été calculées relativement aux deux régions. L'erreur quadratique s'exprime par:

$$EQ = E - E^\wedge = E - \mathbf{u}^T \mathbf{h} = E_1 + E_2 - \mathbf{u}^T \mathbf{h} \quad (4.16)$$

Si l'on n'a pas mémorisé l'expression des énergies des signaux originaux sur chaque région, on peut exprimer l'erreur quadratique par rapport aux erreurs quadratiques EQ_1 et EQ_2 . En effet, pour chaque région, on peut écrire:

$$E_i = EQ_i + E_i^\wedge = EQ_i + \mathbf{u}_i^T \mathbf{h}_i \quad (i=1,2) \quad (4.17)$$

L'expression (4.16) est dès lors remplacée par:

$$EQ = EQ_1 + EQ_2 + \mathbf{u}_1^T \mathbf{h}_1 + \mathbf{u}_2^T \mathbf{h}_2 - \mathbf{u}^T \mathbf{h} \quad (4.18)$$

L'erreur quadratique courante EQC peut être obtenue très facilement en remarquant qu'elle correspond simplement à l'erreur d'approximation qui existe entre les approximations calculées avant et après rassemblement. On peut donc dire que tout se passe comme si l'on prend pour chaque région un signal original ayant exactement la même expression que son approximation au sens des moindres carrés calculée relativement à la région considérée. On a donc la relation $E_i = E_i^\wedge = \mathbf{u}_i^T \mathbf{h}_i$. En remplaçant cette expression dans (4.16), on obtient:

$$EQC = \mathbf{u}_1^T \mathbf{h}_1 + \mathbf{u}_2^T \mathbf{h}_2 - \mathbf{u}^T \mathbf{h} \quad (4.19)$$

On obtient la valeur de l'erreur quadratique moyenne (courante) $EQM(C)$ en divisant $EQ(C)$ par $N=N_1+N_2$, le nombre total de points des deux régions.

En ce qui concerne l'erreur quadratique pondérée EQP, le calcul est un peu plus compliqué. Il faut séparément estimer les différences d'énergies apparaissant sur chaque région. L'erreur commise sur chaque région notée $EQ_{(i)}$ ($i=1,2$) s'exprime par:

$$\begin{aligned}
 EQ_{(i)} &= (\mathbf{g}_i - \mathbf{g}_i^{\wedge})^T (\mathbf{g}_i - \mathbf{g}_i^{\wedge}) \\
 &= E_i - 2 \mathbf{g}_i^{\wedge T} \mathbf{g}_i + \mathbf{g}_i^{\wedge T} \mathbf{g}_i^{\wedge} \\
 &= E_i - 2 (\mathbf{Z}_i \mathbf{u})^T \mathbf{g}_i + (\mathbf{Z}_i \mathbf{u})^T (\mathbf{Z}_i \mathbf{u}) \\
 &= EQ_i + \mathbf{u}_i^T \mathbf{h}_i - 2 \mathbf{u}^T (\mathbf{Z}_i^T \mathbf{g}_i) + \mathbf{u}^T \mathbf{S}_i \mathbf{u} \\
 &= EQ_i + \mathbf{u}_i^T \mathbf{h}_i - 2 \mathbf{u}^T \mathbf{h}_i + \mathbf{u}^T \mathbf{S}_i \mathbf{u} \\
 &= EQ_i + (\mathbf{u}_i - 2 \mathbf{u})^T \mathbf{h}_i + \mathbf{u}^T \mathbf{S}_i \mathbf{u}
 \end{aligned} \tag{4.20}$$

En pondérant respectivement ces contributions par le nombre respectif de points de chaque région, on trouve finalement ($EQP=EQ_{(1)}/N_1+EQ_{(2)}/N_2$):

$$\begin{aligned}
 EQP &= (EQ_1 + (\mathbf{u}_1 - 2 \mathbf{u})^T \mathbf{h}_1 + \mathbf{u}^T \mathbf{S}_1 \mathbf{u})/N_1 + \\
 &\quad (EQ_2 + (\mathbf{u}_2 - 2 \mathbf{u})^T \mathbf{h}_2 + \mathbf{u}^T \mathbf{S}_2 \mathbf{u})/N_2
 \end{aligned} \tag{4.21}$$

En adoptant un raisonnement analogue à celui de l'erreur quadratique courante, on montre que l'erreur quadratique courante pondérée s'exprime simplement par l'erreur quadratique pondérée avec EQ_1 et EQ_2 mis à zéro. On a ainsi:

$$\begin{aligned}
 EQPC &= ((\mathbf{u}_1 - 2 \mathbf{u})^T \mathbf{h}_1 + \mathbf{u}^T \mathbf{S}_1 \mathbf{u})/N_1 + \\
 &\quad ((\mathbf{u}_2 - 2 \mathbf{u})^T \mathbf{h}_2 + \mathbf{u}^T \mathbf{S}_2 \mathbf{u})/N_2
 \end{aligned} \tag{4.22}$$

De toutes ces expressions, il ressort que le calcul des vecteurs d'approximation de chaque région après rassemblement est inutile, les différents critères de dissimilarité définis pouvant tous être obtenus à partir du vecteur des coefficients d'approximation \mathbf{u} sur cette région.

Après avoir étudié comment évoluait l'approximation de chaque région par rassemblement, on explore la structure de données nécessaire à la réalisation du graphe de contiguïté de régions, plaque tournante de l'algorithme de rassemblement.

4.3 Structure des données et implantation du GCR :

Dans un premier temps, on décrit comment réaliser les différents noeuds du graphe par numérotation de région. Le deuxième paragraphe explicite comment les différentes mesures de dissimilarité sont associées aux branches du graphe. On discute ensuite de la nécessité d'ordonner ces diverses mesures avant de commencer le rassemblement à proprement parler. Le système "brain" est alors défini. La section se termine par une étude de l'évolution d'une telle architecture au cours du processus de rassemblement.

4.3.1 Réalisation des noeuds du GCR:

Par définition, un graphe est constitué d'un ensemble de noeuds n_i reliés entre eux par des branches $b_{i,j}$. Ainsi, la branche $b_{i,j}$ relie les noeuds n_i et n_j . Si l'on associe usuellement différents facteurs de pondération aux noeuds et aux branches d'un graphe, ceux-ci ne sont nécessaires qu'aux branches dans le contexte du GCR. Ce sont les mesures de dissimilarité existant entre toutes paires de régions voisines. Ces régions correspondent aux noeuds du GCR, celles-là aux facteurs de pondération des branches qui les lient. Par contre, il semble utile d'associer les paramètres caractéristiques de l'approximation sur chaque région (vecteurs u , h et matrice S) à chaque noeud du graphe. Ceux-ci sont nécessaires au calcul des mesures de dissimilarité ainsi qu'il a été montré à la section précédente.

Une fois l'algorithme de rassemblement terminé, il est nécessaire de reconstruire le signal approximé à l'intérieur de chaque région. Il faut donc qu'à partir de chaque noeud on puisse remonter au domaine de l'image auquel il correspond. Pour ce faire, on peut associer à chaque noeud une liste des points qu'il représente. Il est aussi possible d'associer à chaque noeud une liste des points frontières de la région. Ceci permet de réduire l'espace mémoire nécessaire mais il faut après chaque étape de rassemblement calculer la nouvelle frontière de la région que l'on vient de générer. Dans le premier cas, il suffit de juxtaposer les deux listes de points associées aux deux noeuds que l'on vient de rassembler. De plus, rien n'interdit d'avoir des régions non connexes au cours du rassemblement, ce qui rend le calcul des nouvelles frontières non trivial.

De manière à pouvoir étudier à chaque instant les relations de voisinage entre différentes régions de l'image segmentée, on a choisi d'avoir une représentation mixte graphe-tableau. On associe à chaque noeud du graphe 4 coordonnées représentant les positions de l'extrémité supérieure gauche (k_{\min}, l_{\min}) et inférieure droite (k_{\max}, l_{\max}) du rectangle circonscrit à chaque région. On crée ensuite une image appelée *image de numéros* où chaque pixel prend la valeur n_i du noeud du graphe auquel il appartient. Après chaque étape de rassemblement, on modifie les valeurs de ces pixels en se référant à la position du rectangle circonscrit à la nouvelle région. Celui-ci est défini par la paire de points ($\inf(k_{\min,i}, k_{\min,j}), \inf(l_{\min,i}, l_{\min,j})$) et ($\sup(k_{\max,i}, k_{\max,j}), \sup(l_{\max,i}, l_{\max,j})$), où les indices i et j sont relatifs aux noeuds n_i et n_j impliqués par le rassemblement. L'image de numéros présente l'avantage de définir directement la partition de l'image.

Pour réaliser les noeuds du GCR, on a donc procédé à la création d'une liste allant de 1 à N_r , si la partition de l'image est constituée de N_r régions. Ces valeurs représentent les noeuds du graphe. A chaque noeud sont associés:

- une liste de ses paramètres caractéristiques d'approximation,
- 4 coordonnées correspondant au rectangle circonscrit à la région qu'il représente. Cette région est définie sur une image de numéros.

Analysons maintenant comment réaliser les différentes branches du GCR.

4.3.2 Réalisation des branches du GCR:

A première vue, il suffit d'avoir un tableau dont les indices correspondent aux noeuds du graphe et la valeur prise par le tableau à la mesure de dissimilarité qui existe entre les deux noeuds que la branche relie. S'il n'existe pas de branche entre deux noeuds, la valeur correspondante du tableau est fixée arbitrairement à une valeur très élevée. Ceci est pratiquement irréalisable vu qu'il existe environ 1'000 à 10'000 régions dans le GCR initial. Il faudrait ainsi pouvoir construire un tableau de un million à 100 millions de valeurs.

Une deuxième possibilité consiste à générer une liste de toutes les branches du graphe. A un GCR contenant b branches, on associe une liste de nombres allant de 1 à b . A chacun de ces nombres correspondent trois valeurs. Les deux premières définissent les noeuds n_i et n_j auxquels cette branche se rapporte. La dernière correspond à la mesure de dissimilarité associée. Malheureusement ce choix s'avère en pratique être tout aussi inefficace que le premier car il est nécessaire d'éliminer à chaque étape de rassemblement les branches associées aux deux noeuds que l'on a rassemblés. Il devient dès lors nécessaire de parcourir toute la liste de manière à trouver ces branches.

Pour éviter ce problème, il faut ordonner cette liste en fonction des numéros relatifs aux noeuds. Il devient dès lors nécessaire de répéter chaque branche pour rendre cette classification aisée.

Une solution plus élégante consiste à associer aux N_r éléments de la liste des noeuds, une liste de ses voisins, autrement dit une liste des noeuds auxquels il est relié par une branche du GCR. Chaque voisin définit une cellule. Celle-ci est caractérisée par le numéro du voisin et une valeur correspondant à la mesure de dissimilarité qui existe entre lui et l'élément de la liste des noeuds dont il fait partie. On a ainsi accès à toute l'information du GCR. Chaque fois qu'un rassemblement est réalisé, il suffit de modifier la liste de voisins de tous les noeuds voisins de la nouvelle région. Le noeud obtenu par rassemblement a pour voisins l'union des voisins des deux régions que l'on vient de rassembler. Le calcul de cette réunion peut être optimisé en ordonnant la liste des voisins.

De manière à éviter de renuméroter tous les noeuds à chaque étape de rassemblement, on décide d'associer au nouveau noeud la valeur du plus petit numéro des deux noeuds que l'on vient de rassembler.

Pour diminuer l'espace mémoire nécessaire, on ne définit une cellule que dans le cas où le voisin porte un numéro plus élevé que le noeud auquel il se

rapporte. Dans les autres cas, la cellule correspondant au voisin v d'un noeud n_i est remplacée par un pointeur appelé *pointeur de cellule* qui indique la cellule correspondant au voisin n_i associé au noeud v de la liste des noeuds.

Comme il existe de multiples branches dans un GCR surtout au début du rassemblement, il est essentiel de pouvoir accéder rapidement à la plus faible mesure de dissimilarité ainsi qu'à toutes celles associées aux branches où l'on vient d'effectuer un rassemblement.

4.3.3 Mise à jour et ordonnancement des mesures de dissimilarité:

La modification du GCR intervient toujours à l'endroit de dissimilarité minimale. Comme le nombre de branches du graphe est considérable, elles vont être triées pour pouvoir accéder rapidement au minimum de dissimilarité. Après chaque étape de rassemblement, il est nécessaire d'éliminer les branches qui ont disparu et de recalculer les mesures de dissimilarité relatives aux branches qui lient le nouveau noeud à ses voisins. Le nombre de voisins d'un certain noeud est très dépendant de la forme géométrique des régions auxquelles lui et ses voisins correspondent. Il est donc indispensable d'avoir une structure dynamique. L'ordonnancement des mesures de dissimilarité peut être obtenu au moyen d'une structure dynamique qui lie les différentes branches du graphe.

Deux possibilités peuvent être utilisées. La première consiste à pointer pour chaque noeud de la liste de noeuds sur la branche qui présente la dissimilarité minimale, c'est-à-dire la cellule présentant le minimum de dissimilarité dans la liste des voisins du noeud considéré. Il suffit de trier dès lors au plus $N_f - 1$ mesures de dissimilarité au lieu de b . La seconde consiste à définir une structure d'*arbre binaire balancé* entre les différentes mesures de dissimilarité.

Un arbre est dit balancé [59] si et seulement si, pour chaque noeud de cet arbre les hauteurs de ses sous-arbres diffèrent au plus de 1. Les opérations suivantes peuvent être réalisées en au plus $O(\log b)$ opérations si b correspond au nombre d'éléments à trier:

- trouver un noeud de l'arbre possédant une certaine valeur,
- insérer dans l'arbre un noeud possédant une certaine valeur,
- éliminer de l'arbre un noeud possédant une certaine valeur.

Un algorithme permettant d'insérer un élément dans l'arbre avec rebalancement de ce dernier dépend fortement de la façon dont est mémorisée l'information de balancement. Celle-ci peut faire partie intégrante de chaque noeud de l'arbre. il est nécessaire de la mettre à jour à chaque étape d'insertion ou d'effacement d'un élément de l'arbre. Une fois que l'on a déterminé où insérer un nouvel élément, on pratique les opérations de rebalancement par permutation de pointeurs. Ceux-ci sont échangés cycliquement en une simple ou double rotation de deux ou trois noeuds à chaque niveau de l'arbre. L'information de balancement est également modifiée en accord avec ces permutations cycliques.

Chaque étape de rassemblement entre deux noeuds n_i et n_j entraîne la modification de l'arbre balancé. La branche liant n_i à n_j est supprimée. Si le nouveau noeud porte le numéro n_i , toutes les branches liant n_j à des voisins de n_i avant rassemblement sont également éliminées. Les valeurs associées aux branches restantes sont modifiées de manière à correspondre aux mesures de dissimilarité entre le nouveau noeud et ses voisins. Le nouvel arbre balancé est obtenu en détruisant un nombre de cellules identique au nombre de voisins communs à n_i et à n_j , et en rebalançant l'arbre pour les autres mesures de dissimilarité excepté celle qui était associée à la branche liant n_i à n_j . S'il existe initialement b_i branches associées à n_i et b_j branches associées à n_j , la modification de l'arbre balancé peut être réalisée en $O((b_i+b_j) \log(b))$ opérations, où b représente le nombre total de branches du GCR.

On a adopté une représentation du GCR où l'ensemble des noeuds définissent une liste ordonnée. Outre les informations relatives à la forme et à l'approximation de chaque région, on associe à chaque noeud une liste de voisins représentant les numéros de ses voisins et les mesures de dissimilarité correspondantes. Celles-ci sont ordonnées sur la base des mesures de dissimilarité au moyen d'un arbre binaire balancé. On accède au minimum de dissimilarité en $O(\log(b))$ opérations par parcours de l'arbre binaire balancé. La cellule associée fait apparaître l'un des noeuds impliqués dans le rassemblement. Pour pouvoir connaître l'autre noeud instantanément, on associe encore à chaque cellule un nombre correspondant au numéro du noeud auquel cette cellule se rapporte. On aboutit ainsi à la forme définitive de la structure *Brain*.

4.3.4 Structure BRAIN:

Le schéma de cette structure est illustré à la figure 4.3. La partie relative à l'arbre balancé est présentée plus en détail à la figure 4.4.

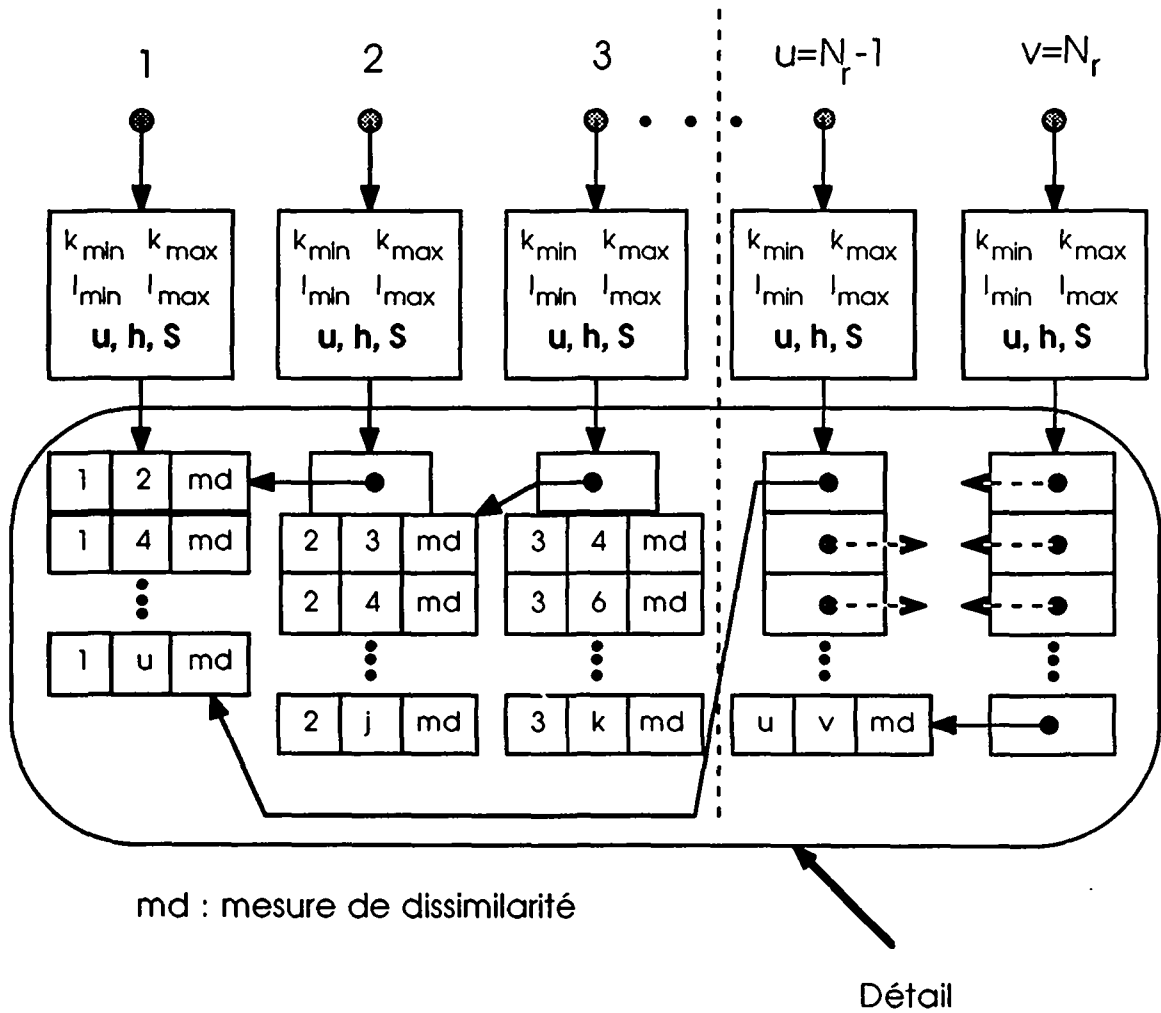


Figure 4.3: Structure Brain.

Chaque fois qu'un voisin porte un numéro moins élevé que le noeud auquel il se rapporte, le voisin correspondant (et la mesure de dissimilarité qui lui est associée) sont obtenus en suivant le pointeur de cellule. La cellule pointée possède deux cases où voisin et noeud sont inversés. Ainsi la mesure de dissimilarité correspondant à la branche 2-1 apparaît à la cellule 1-2-md.



Figure 4.4: Détail de l'arbre équilibré

Après chaque étape de rassemblement, la structure Brain évolue en ce que le noeud portant le numéro le plus élevé disparaît. L'autre noeud correspond à la nouvelle région après rassemblement. Ses paramètres d'approximation et de forme de région (y compris la partie de l'image de numéro associée) sont mis à jour et sa liste de voisins s'allonge des cellules qu'il a acquises par rassemblement. Selon la valeur du nouveau numéro, les cellules doivent être modifiées voire remplacées par des pointeurs de cellules. Ceci est décrit dans l'annexe F. En dernière étape, on rebalance l'arbre en fonction des nouvelles mesures de dissimilarité. Le choix d'une telle structure permet non seulement d'accéder rapidement à la plus faible mesure de dissimilarité mais il assure l'accès immédiat à toutes les mesures de dissimilarité et voisins de chaque région. L'ordonnancement des cellules par rapport à la valeur du voisin est réalisé au moyen d'une liste pointée. Cette autre structure garantit un calcul rapide de l'union des deux listes de voisins (voir annexe F).

Avec une telle architecture, il s'agit maintenant d'estimer la complexité de l'algorithme de rassemblement pour aboutir à la segmentation finale.

4.4 Complexité de calcul et solutions sous-optimales :

Cette section met en évidence la complexité de l'algorithme si l'on utilise la structure de données Brain pour représenter le GCR. Le premier paragraphe décrit le nombre d'opérations nécessaires à la construction du GCR initial et à chaque étape de rassemblement. On étudie ensuite comment il est possible de simplifier les calculs liés à l'approximation et aux mesures de dissimilarité lorsqu'on analyse le rassemblement potentiel d'une région de très grande taille avec une région de très petite taille (par exemple un point image). Cette analyse est suivie de la proposition de méthodes sous-optimales pour effectuer le rassemblement mais qui garantissent une qualité comparable.

4.4.1 Complexité de l'algorithme:

Il faut distinguer deux catégories de traitements. Le premier est relatif à l'élaboration de la structure Brain de départ à partir de la partition de l'image obtenue par l'algorithme de division. Le second correspond au nombre d'opérations nécessaires à chaque étape de rassemblement. Par la suite, on supposera que le GCR initial contient b branches et N_r noeuds. On admettra également d'utiliser un ensemble de r fonctions de base pour approximer les données sur chaque région. Pour des polynômes de degré 0, 1, 2 ou 3, on aura respectivement $r=1, 3, 6$ ou 10 .

Pour obtenir la structure Brain initiale, il faut dans un premier temps générer la liste des N_r noeuds et les caractéristiques d'approximation et de forme de région relatives à chacun de ces noeuds. On ne peut pas utiliser directement les expressions données par l'algorithme de division, vu qu'il est nécessaire d'exprimer ces paramètres d'approximation pour un choix d'origine unique (x_0, y_0) pour pouvoir appliquer la propriété de décomposition de l'approximation au sens des moindres carrés (voir paragraphe 4.2.2). On peut éviter les calculs de l'approximation sur chaque carré après changement d'origine en appliquant une simple matrice de changement de base C aux paramètres de l'approximation déjà calculés. C est de dimension $r \times r$ et est définie par la relation:

$$\Psi'(x, y) = C \Psi(x - x_0, y - y_0) \quad (4.22)$$

Les nouveaux paramètres de l'approximation sont donnés par les relations (voir annexe G) :

$$S' = C S C^T \quad (4.23)$$

$$u' = (C^T)^{-1} u \quad (4.24)$$

$$h' = C h \quad (4.25)$$

Ces expressions ne font nullement intervenir le nombre de points de chaque région. Pour chaque noeud du graphe, ces calculs requièrent environ $O(r^2)$ opérations. Les paramètres de forme de région sont obtenus en numérotant une image de numéros en fonction de la structure de l'image après application de l'algorithme de division. Les carrés contenant des contours sont mis à zéro dans l'image des numéros. Ils seront traités ultérieurement (voir paragraphe 4.1.1). Pour

chaque noeud du GCR, il s'agit maintenant d'estimer l'ensemble de ces voisins. La liste ordonnée des voisins est obtenue en parcourant l'image de numéros. Chaque fois que deux pixels consécutifs non nuls de cette image selon une certaine ligne ou une certaine colonne ont une valeur différente, on se trouve à la frontière entre deux carrés. On teste dans le carré de numéro le plus faible s'il existe la cellule correspondant à l'autre carré dans la liste de ses voisins. Si tel est le cas on passe au pixel suivant, sans quoi cette cellule est insérée dans la liste des voisins et un pointeur de cellule associé est inséré dans la liste des voisins du carré portant le numéro le plus élevé. La génération d'une telle liste requiert $O(N^2)$ comparaisons pour une image de taille $N \times N$. Pour compléter la liste des opérations à la génération du Brain initial, il s'agit de parcourir les b cellules de cette structure correspondant aux branches du GCR et de leur associer une mesure de dissimilarité. Le calcul de ces b mesures de dissimilarité est réalisé en $O(br^3)$ opérations quelle que soit le critère choisi. Le calcul diffère d'un critère à l'autre d'environ $2r^2$ opérations par branche si l'on estime l'erreur quadratique pondérée (courante) (2 multiplications matrice-vecteur de dimension r supplémentaires). Pour tous les autres critères, le gros des opérations correspond essentiellement à l'inversion de la matrice S associée à l'ensemble des deux régions impliquées dans le rassemblement. Une fois l'ensemble des mesures de dissimilarité connues, celles-ci sont triées au moyen d'un arbre binaire balancé en $O(b \log(b))$ opérations.

En comparant les différentes phases de la génération du Brain initial, on remarque que pour des approximations polynomiales de degré élevé, le calcul des mesures de dissimilarité s'avère la tâche la plus coûteuse vu que r^3 est beaucoup plus grand que $\log(b)$. Lorsque le degré est plus faible, c'est le tri de ces mesures qui requiert le plus grand nombre d'opérations. On remarque cependant qu'il aurait été pratiquement impossible de réaliser l'algorithme de rassemblement pour des polynômes de degré élevé si l'approximation au sens des moindres carrés ne bénéficiait pas de la propriété de décomposition.

Notons qu'une modification du choix du critère de dissimilarité au cours du processus de rassemblement n'entraîne pas une modification de la structure Brain mais nécessite la réévaluation de toutes les mesures de dissimilarité et leur triage au moyen d'un arbre binaire balancé.

Estimer la complexité de calcul de tout le processus de rassemblement est très difficile, car celle-ci est fortement dépendante de l'évolution de la segmentation et donc de l'information contenue dans l'image analysée. Aussi, on s'est borné à étudier quelles étaient les tâches les plus coûteuses lors de chaque étape de rassemblement.

Après rassemblement, il faut modifier la structure Brain de la façon suivante:

- estimer les nouveaux paramètres de forme (rectangle circonscrit) et d'approximation (matrice S , vecteurs u et h) de la région obtenue par rassemblement. Ceci requiert seulement $O(r^3)$ opérations.

- modifier la liste des voisins du noeud n_i portant le numéro le plus bas. Ceci est obtenu par insertion des voisins de l'autre noeud n_j qui n'étaient pas les siens mis à part n_i . Si le noeud n_j a b_j branches, la liste de voisins peut être modifiée en $O(b_j)$ comparaisons du fait de l'ordonnancement de cette liste.

- Si après rassemblement n_i possède b_i' voisins, il s'agit d'estimer les b_i' nouvelles mesures de dissimilarité, ce qui est effectué en environ $O(b_i' r^3)$ opérations.

- Enfin, il faut rebalancer l'arbre qui lie les différentes mesures de dissimilarité. S'il y avait b_i branches associées à n_i avant rassemblement, le rebalancement de l'arbre est réalisé en $O((b_i + b_i') \log(b))$ opérations (voir paragraphe 4.3.3) où b représente la moyenne du nombre de branches entre le GCR avant et le GCR après rassemblement. Une fois l'arbre rebalancé, le calcul du minimum de dissimilarité est réalisé en $O(\log(b'))$ opérations si b' correspond au nombre de branches dont est formé le GCR après l'étape de rassemblement.

A chaque étape de rassemblement, il faut également donner aux pixels de valeur n_j la valeur n_i , dans l'image de numéros. Ceci est réalisé en $(k_{\max,j} - k_{\min,j} - 1) \times (l_{\max,j} - l_{\min,j} - 1)$ comparaisons.

De tout ce qui précède, on remarque que les opérations liées au calcul et à l'ordonnancement des mesures de dissimilarité sont les plus coûteuses. Si le degré de l'approximation est élevé, le calcul des mesures de dissimilarité est le plus long; dans un cas contraire, l'ordonnancement des mesures de dissimilarité nécessite le plus de temps calcul.

La complexité de calcul peut varier beaucoup d'une étape de rassemblement à l'autre. Tout dépend du nombre de voisins de chaque région impliquée dans le rassemblement.

Dans ce qui suit, des méthodes sous-optimales liées soit à la structure de l'algorithme, soit au calcul de l'approximation sont proposées pour accélérer le processus de rassemblement.

4.4.2 Simplification du calcul d'approximation:

Même si l'on peut efficacement utiliser la propriété de décomposition de l'approximation selon le critère des moindres carrés, on remarque que les calculs liés à l'approximation (mesures de dissimilarité) définissent la part la plus coûteuse de l'algorithme de rassemblement.

On peut les simplifier davantage dans le cas des rassemblements impliquant des régions de tailles très disproportionnées. En effet, dans de tels cas, on peut utiliser la théorie des perturbations.

Il est raisonnable d'admettre que l'expression de la matrice S après rassemblement sera très proche de celle de la région de très grande taille, correspondant au domaine D_1 . On a donc:

$$S \equiv S_1 \quad (4.26)$$

L'expression (4.7) se simplifie dès lors de la façon suivante:

$$u \equiv (S_1)^{-1} (S_1 u_1 + S_2 u_2) = u_1 + S_1^{-1} S_2 u_2 = u_1 + S_1^{-1} h_2 \quad (4.27)$$

Si S_1^{-1} a été mémorisé, on réduit la complexité de calcul par rapport à (4.7) de $O(r^3)$ à $O(r^2)$. De la même manière, les calculs peuvent être simplifiés pour le calcul des mesures de dissimilarité. En faisant l'hypothèse que (4.26) reste applicable, la relation (4.18) peut être utilisée avec u calculé selon (4.27). En ce qui concerne l'erreur quadratique pondérée, on peut négliger la contribution de l'erreur sur le premier domaine due à la modification de l'approximation, car celle-ci reste pratiquement inchangée du fait du grand nombre de points de ce domaine. On a alors:

$$EQP \equiv EQ_1/N_1 + (EQ_2 + (u_2 - 2u)^T h_2 + u^T S_2 u)/N_2 \quad (4.28)$$

De la même façon, l'erreur quadratique moyenne est obtenue directement en pondérant l'erreur quadratique par le nombre N_1 de points du domaine D_1 . Pour obtenir les expressions des critères de dissimilarité courants, on applique les expressions (4.18) ou (4.28) en posant $EQ_1=EQ_2=0$. Le calcul des mesures de dissimilarité a été ainsi réduit de $O(r^3)$ opérations à $O(r^2)$ opérations.

Il faut cependant souligner que ces approximations ne sont pas applicables si elles sont utilisées de manière répétée à une même grande région de référence. A partir d'un certain nombre de rassemblements, la contribution des petites régions n'est plus négligeable. Les équations (4.27) et (4.28) s'écartent alors de la réalité. Il devient nécessaire de corriger les estimations effectuées après un certain nombre de rassemblements en recalculant l'expression exacte des paramètres d'approximation au moyen des relations (4.9) à (4.11). Aussi est-il impératif de mémoriser les paramètres relatifs à toutes les régions impliquées par la simplification. Le nombre de rassemblements est choisi en liaison avec la somme du nombre de points de toutes les petites régions qui ont été insérées à la grande. Dès que ce nombre n'est plus négligeable par rapport au nombre initial de points de la grande région, on corrige les estimations des paramètres d'approximation.

Lorsque des régions de la taille d'un pixel sont associées à de plus grandes régions, les relations (4.27) et (4.28) restent parfaitement valables. Si $g(k,l)$ est la valeur du pixel, le vecteur h_2 correspondant s'exprime par le vecteur de dimension r $g(k,l)\Psi(k,l)$. De même, la matrice S_2 s'écrit alors $\Psi(k,l)\Psi^T(k,l)$.

4.4.3 Solutions sous-optimales par modification du processus de rassemblement:

Une autre possibilité permettant de réduire la complexité algorithmique du processus de rassemblement est obtenue en modifiant légèrement son principe.

Ainsi, on peut par exemple recalculer les mesures de dissimilarité seulement après un certain nombre de rassemblements. Il devient alors simplement nécessaire d'interdire le rassemblement de toutes les régions qui ont été impliquées précédemment dans un rassemblement sans que les mesures de dissimilarité n'aient été corrigées. Cette procédure n'introduit guère de distorsion appréciable au début du rassemblement. Elle évite notamment le balancement de l'arbre binaire balancé à chaque étape de rassemblement, qui est justement coûteux lorsque le GCR possède beaucoup de branches. Plus le nombre de

régions diminue, plus il devient nécessaire de corriger souvent les mesures de dissimilarité [60].

Il est également possible de limiter la taille maximale du GCR en nombre de noeuds. A partir de la partition initiale de l'image par l'algorithme de division, on construit un GCR comprenant les $N(<N_r)$ premières régions de cette segmentation. Tant qu'il existe des régions de la partition initiale non encore analysées dans le GCR, celles-ci sont insérées une à une après chaque étape de rassemblement. On réduit ainsi le nombre de comparaisons des mesures de dissimilarité tout en garantissant la correspondance cherchée entre les objets de l'image et la forme de la segmentation. En effet, il est peu probable que ce qui se passe dans un coin de l'image ait une influence immédiate sur ce qui se passe dans un autre coin.

Si le principe de rassemblement a été modifié dans les derniers paragraphes pour en accélérer le traitement, les résultats présentés dans la suite de notre travail correspondront toujours à l'application de l'algorithme de rassemblement sans simplification.

Si l'on vient d'étudier la complexité de calcul de l'algorithme de rassemblement, on analyse dans la section suivante la stratégie à adopter dans le choix des mesures de dissimilarité pour obtenir la segmentation finale de l'image.

4.5 Stratégie globale de rassemblement :

Dans cette partie du chapitre 4, on étudie dans un premier temps les erreurs dues au choix des mesures de dissimilarité d'une part, au fait qu'il existe plus d'un paramètre d'approximation d'autre part. Cette analyse justifiera de procéder comme suit:

1) rassemblement des régions sans contours, en appliquant successivement l'erreur quadratique pondérée courante et l'erreur quadratique courante comme mesure de dissimilarité.

2) insertion des régions situées de part et d'autre des contours dans le GCR. Ces régions sont obtenues avec l'aide de l'image de contrôle des structures positives et négatives (voir section 2.2).

3) rassemblement adaptatif par erreur quadratique courante jusqu'à obtenir le nombre de régions désiré de la segmentation finale.

4.5.1 Analyse des défauts:

Les défauts de la procédure de rassemblement peuvent provenir d'un mauvais choix du critère de dissimilarité et de l'impossibilité de pouvoir générer des zones homogènes de petite taille situées proches des contours d'une image à cause de l'augmentation du degré du polynôme d'approximation.

Voyons dans une premier temps les effets des différents critères de dissimilarité.

4.5.1.1 Analyse expérimentale des effets des critères de dissimilarité:

Si l'algorithme de rassemblement proposé est optimal en ce que ce rassemblement a lieu à chaque étape à l'endroit de dissimilarité minimale, rien ne garantit de préserver l'information contour de l'image dans la segmentation finale.

Comme il a été mentionné à la section 4.1, chaque critère de dissimilarité présente un certain nombre d'avantages et d'inconvénients. Selon l'état du GCR, un critère peut s'avérer plus efficace qu'un autre. Il s'agit donc de proposer comment faire évoluer les différentes mesures de dissimilarité de manière à tenir compte au mieux de la structure des régions.

On va montrer expérimentalement les effets des différents critères dans ce qui suit. Deux hypothèses vont permettre de définir la modalité de l'expérimentation:

- le comportement d'un certain critère quant à la structure de la segmentation est peu dépendant du degré du polynôme d'approximation. Pour simplifier le test, on utilisera donc des polynômes de degré 0. En effet, si le résultat de la segmentation finale est très dépendant du choix de l'approximation, l'effet de chaque critère de dissimilarité quant à sa tendance à agglomérer des petites régions à des grandes ou à favoriser le regroupement de petites régions est indépendant de l'approximation choisie.

- l'analyse des effets de chaque critère de dissimilarité peut être faite indépendamment de la taille de l'image analysée. En outre, il est raisonnable d'admettre que les résultats obtenus à partir d'une seule image naturelle permettent déjà de bien comprendre le comportement d'un critère de dissimilarité, vu que l'on a à faire à un très grand nombre de configurations possibles de formes et tailles de régions tout au long du processus de rassemblement. Aussi avons-nous décidé de prendre l'image du couple pour effectuer notre expérience.

Pour mettre en évidence le comportement de chaque critère, on définit l'expérience suivante. A partir de l'image du couple segmentée par l'algorithme de division au moyen d'une représentation par polynômes de degré 0, on va étudier le comportement de cinq critères de dissimilarité tout au long du processus de rassemblement. La segmentation donnée par l'algorithme de division est présentée à la figure 3.25b. De manière à limiter le nombre de régions du GCR, on a associé les pixels situés de part et d'autre des contours de l'image donnés par l'image de contrôle des contours (fig. 2.15d) sur la base de l'image des structures positives et négatives de l'image du couple (fig. 2.24a). Le GCR initial comprend dès lors 5596 noeuds.

Au cours de l'expérience, on va observer la structure de la segmentation et de l'image reconstruite correspondante à six instants de la procédure de rassemblement. Ces instants sont définis en relation avec le nombre de régions qui constituent la segmentation de l'image. Dans notre cas, on étudie la forme de la partition pour 4999, 2999, 999, 199, 99 et 49 régions.

La forme de la segmentation est décrite en assignant à chaque région un niveau de gris parmi quatre. On tire profit du théorème des quatre couleurs de la théorie des graphes [61] qui garantit qu'il est possible de colorier une carte de pays au moyen de quatre couleurs seulement pour autant qu'il n'existe pas plus de quatre pays voisins en un seul point. En réalité, pour des raisons de simplicité de

programmation, on a eu recours à une représentation au moyen de cinq niveaux de gris. Ainsi, tous les points d'une région portent un certain niveau de gris, sans qu'il soit possible d'avoir deux régions voisines dont les points respectifs possèdent la même valeur.

Pour un niveau de rassemblement donné, on présente successivement les résultats de rassemblement obtenus au moyen de l'erreur quadratique moyenne (EQM), de l'erreur quadratique (EQ), de l'erreur quadratique courante (EQC), de l'erreur quadratique courante pondérée (EQCP) et de l'erreur quadratique pondérée (EQP). L'erreur quadratique moyenne courante n'a pas été insérée dans ce banc de test vu les résultats peu performants que l'on obtient avec l'erreur quadratique moyenne (comme on le verra), ces deux mesures ayant un comportement similaire.

Les sous-numéros a, b, c, d et e de chaque figure correspondent aux critères susmentionnés dans l'ordre dans lequel ils ont été présentés.

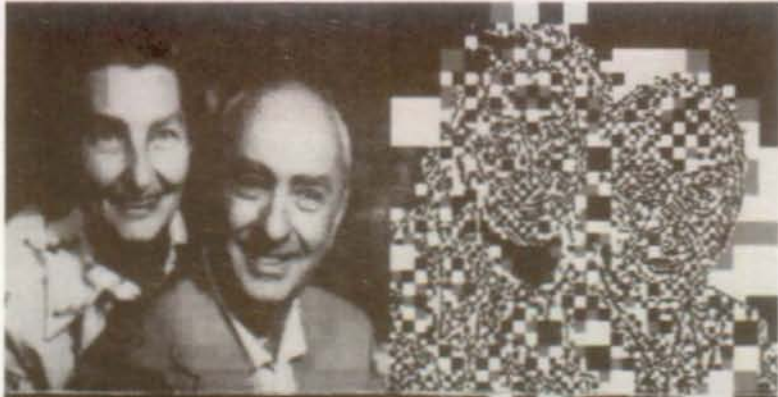
La figure 4.5 présente les résultats d'une segmentation à 4999 régions. Après environ 600 rassemblements (de 5596 à 4999), il est difficile de distinguer les résultats donnés par chaque critère. Sur le plan visuel, aucune différence n'apparaît au niveau de l'image reconstruite. On remarque cependant déjà que les critères EQ et EQC (figures 4.5b et 4.5c) ont tendance à favoriser le regroupement de petites régions ressemblantes alors que les autres critères EQM, EQCP et EQP (figures 4.5a, 4.5d et 4.5e) tendent à rassembler des régions de grande taille d'abord. Il suffit d'observer pour cela la forme des régions correspondant à l'arrière-plan sombre de l'image du couple. Dans toutes les images de la figure 4.5, une structure de géométrie rectangulaire est imposée à chaque région à cause de la forme de la segmentation initiale obtenue par l'algorithme de division.

La figure 4.6 présente les résultats d'une segmentation à 2999 régions. Les remarques effectuées pour la figure 4.5 sont confirmées pour cette figure. Les résultats donnés par les critères EQM, EQCP et EQP (figures 4.6a, 4.6d et 4.6e) semblent plus attrayants à ce niveau du processus de rassemblement vu que l'on voit apparaître des grandes régions correspondant à de larges zones homogènes de l'image analysée (par exemple, arrière-plan ou cou de la femme sur l'image du couple). À ce niveau de rassemblement, il est pratiquement impossible de déceler une distorsion plus grande que celle que l'on avait au moment de la segmentation initiale (après l'application de l'algorithme de division). La segmentation la plus correcte semble être donnée ici par le critère d'erreur quadratique pondérée. En effet, c'est le seul critère pour lequel on a des régions uniformes de très grande taille correspondant à des objets réels de la scène et pour lequel la structure des frontières à angle droit, imposée par l'algorithme de division commence à disparaître.

La figure 4.7 présente les résultats d'une segmentation à 999 régions. Le comportement décrit précédemment pour les différents critères se confirme: rassemblement des petites régions entre elles pour EQ et EQC (figures 4.7b et 4.7c); rassemblement de régions de grande taille pour les autres critères étudiés. On voit cependant apparaître les premiers désavantages qui entraînent des distorsions au niveau de l'image reconstituée. En ce qui concerne l'erreur quadratique moyenne, une région ayant atteint une taille importante comme l'arrière-plan de l'image du couple a tendance à avaler des régions de très petite taille qui appartiennent néanmoins à d'autres espaces réguliers de l'image. Ceci

peut être observé à la figure 4.7a au niveau de l'épaule de l'homme et au voisinage de l'oreille gauche de la femme. Cette distorsion est inacceptable et elle justifiera le rejet de cette mesure de dissimilarité pour comparer les régions lors du processus de rassemblement. De plus, EQM ne permet pas nécessairement d'obtenir des régions couvrant des espaces réguliers d'aussi grande taille que les critères EQCP et EQP. Si des grandes parties d'espaces réguliers sont ainsi obtenues avec les critères EQCP et EQP (figures 4.7d et 4.7e), on commence à avoir des régions de très petite taille ne représentant pas d'information pertinente, placées comme des enclaves dans de très grandes régions. Observer à cet effet, les micro-régions apparaissant au bas de l'épaule gauche de l'homme. En ce qui concerne les critères EQ et EQC (figures 4.7b et 4.7c), on remarque un certain nombre d'avantages par contre: il n'existe plus de régions de très petite taille. On a ainsi éliminé les détails de l'image, sans que le gros de l'information sémantique ait été perdu. Cette considération s'avère particulièrement intéressante dans le contexte du codage. Mis à part l'erreur quadratique moyenne, la qualité des différentes images est comparable à ce niveau de la segmentation.

a



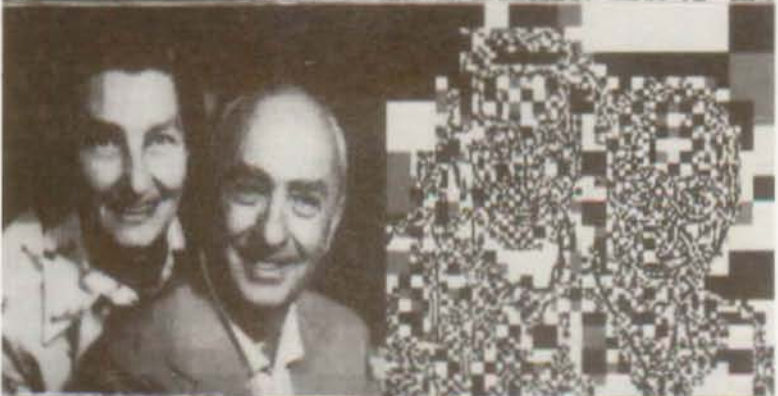
b



c



d



e

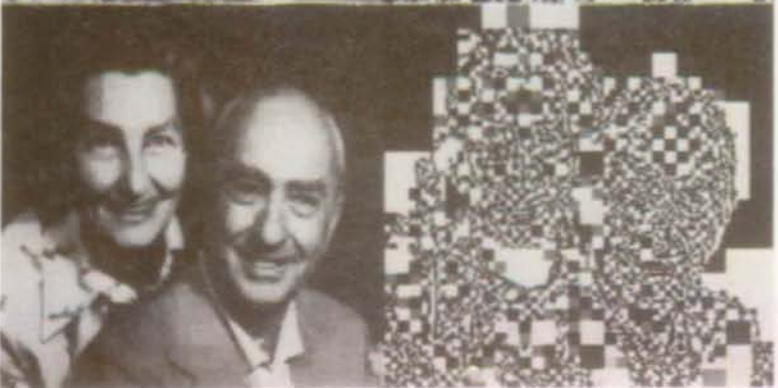
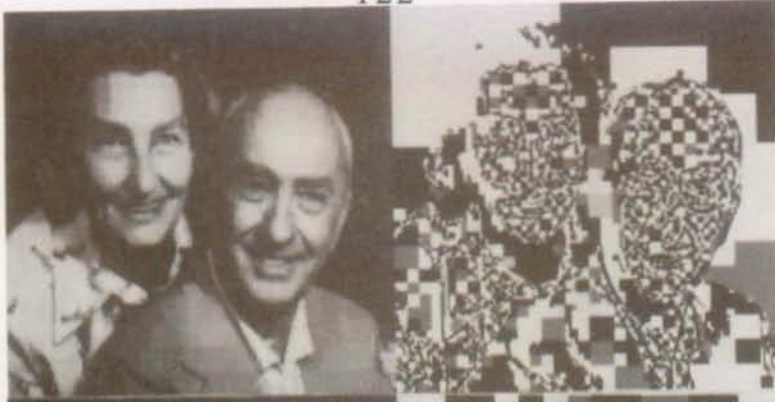


Figure 4.5: Segmentation formée de 4999 régions.
(ordre de présentation : EQM,EQ,EQC,EQCP,EQP)

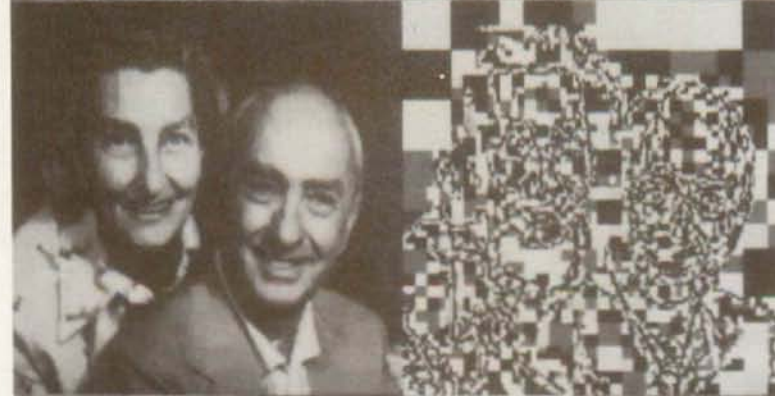
a



b



c



d



e

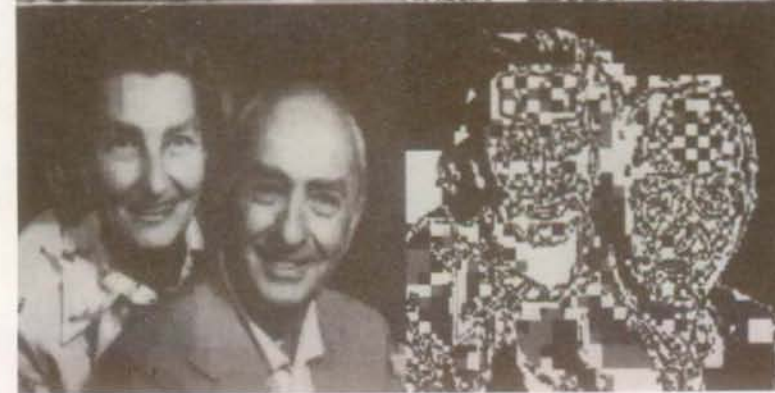


Figure 4.6: Segmentation formée de 2999 régions.
(ordre de présentation : EQM,EQ,EQC,EQCP,EQP)

a



b



c



d



e



Figure 4.7: Segmentation formée de 999 régions.
(ordre de présentation : EQM,EQ,EQC,EQCP,EQP)

La figure 4.8 montre les résultats d'une segmentation à 199 régions. On n'a pas présenté de segmentations constituées d'un nombre de régions compris entre 999 et 199, car aucun autre type de distorsion n'était visible. Les phénomènes décrits jusque là pour caractériser chaque critère restent parfaitement valables. En observant les structures des segmentations obtenues avec chaque critère, on commence à voir apparaître la structure des frontières des espaces réguliers de l'image, comme les yeux ou les dents des personnages, leur tête... Les désavantages occasionnés par le critère d'erreur quadratique moyenne (figure 4.8a) sont encore plus visibles à ce niveau. La plupart des frontières réelles des objets de l'image sont déformées (dimension de la tête des personnages). Si l'on compare l'image de la figure 4.8a aux images des figures 4.8b à 4.8e, on a une considérable baisse de qualité. En comparant les figures 4.8b et 4.8c aux figures 4.8d et 4.8e, on remarque que les segmentations obtenues avec EQ et EQC présentent des zones homogènes de grande taille subdivisées en plusieurs régions (arrière-plan de l'image du couple, front des personnages...) alors que celles obtenues à partir de EQCP et EQP ont des régions de très petite taille correspondant à des détails insignifiants non incluses dans des régions avoisinantes. Ces défauts sont complémentaires. En inversant ces deux types de critères de dissimilarité (pondéré et non pondéré), on peut associer les toutes petites régions aux grandes pour les segmentations correspondant aux figures 4.8d et 4.8e ou associer des grandes régions définissant un même espace régulier pour les segmentations correspondant aux figures 4.8b et 4.8c. Si tel n'est pas le cas des distorsions inacceptables commencent à apparaître en poursuivant le processus de rassemblement. Ceci est déjà visible pour les figures 4.8d et 4.8e sur la partie droite du front de l'homme.

Dans les figures 4.9 et 4.10 représentant chacune des segmentations de 99 et 49 régions, on met mieux en évidence ces distorsions. Sans parler de l'erreur quadratique moyenne, on remarque la qualité dérisoire des images obtenues avec EQCP et EQP (Figures 4.9d, 4.9e, 4.10d et 4.10e). Pour ce qui est des critères EQ et EQC, la qualité est nettement plus élevée, si ce n'est que l'arrière-plan de l'image est constitué de trois régions pour la figure 4.9, de deux pour la figure 4.10. Vu la taille de ces régions, la longueur des frontières correspondant à des faux contours est importante. Leur codage est en conséquence très coûteux, ce qui réduit la compression des figures 4.9b, 4.9c et 4.10b, 4.10c par rapport à celle des figures 4.9d, 4.9e et 4.10d, 4.10e, respectivement. Il s'agit donc d'utiliser une stratégie basée sur les critères EQ et EQC où l'on ait pris cependant le soin d'appliquer pendant quelques rassemblements au début de la procédure les critères EQCP et EQP pour regrouper les régions correspondant à de très grands espaces réguliers, régions qui ne pourraient être rassemblées si l'on choisissait les critères EQ ou EQC. Parmi EQ et EQC, on peut remarquer en comparant les figures 4.10b et 4.10c, que l'erreur quadratique courante permet d'obtenir une meilleure précision (tous les yeux des personnages sont conservés). Les raisons de ceci sont difficiles à déterminer, si ce n'est que EQC permet de mesurer exactement la distorsion introduite par le rassemblement de deux régions pour le signal d'approximation calculé sur ces deux régions, autrement dit pour le modèle qui caractérise l'homogénéité de ces régions (voir paragraphe 4.1.2). S'il n'y a pas eu de distorsion significative de l'image quant à son contenu sémantique, EQC s'avère donc une mesure plus adéquate.

a



b



c



d



e



Figure 4.8: Segmentation formée de 199 régions.
(ordre de présentation : EQM,EQ,EQC,EQCP,EQP)

a



b



c



d



e



Figure 4.9: Segmentation formée de 99 régions.
(ordre de présentation : EQM,EQ,EQC,EQCP,EQP)

a



b



c



d



e



Figure 4.10: Segmentation formée de 49 régions.
(ordre de présentation : EQM,EQ,EQC,EQCP,EQP)

4.5.1.2 Distorsions causées par l'augmentation du degré du polynôme :

Si l'augmentation du degré du polynôme permet de définir des régions caractérisées par une variation complexe du signal de luminance, certains artefacts peuvent être introduits à cause de la singularité de la matrice \mathbf{S} dans le calcul de l'approximation selon le critère des moindres carrés.

La précision imposée aux frontières des régions correspondant à des contours réels de l'image exige de subdiviser les carrés qui contiennent des contours, en l'ensemble des points qui les définissent (voir paragraphe 4.1.1). Ces points sont insérés dans le GCR comme noeuds à part entière. Deux désavantages résultent d'une telle solution. D'une part, on augmente ainsi considérablement le nombre de noeuds du GCR, nécessitant un temps énorme de traitement. D'autre part, il est impossible d'avoir des zones homogènes allongées situées sur une bande étroite de part et d'autre d'un contour (par exemple, les sourcils ou les dents d'un individu). La figure 4.11 montre comment l'ellipse en noir a disparu une fois que l'image a été segmentée. Ce genre de distorsion est d'autant plus vrai que le degré du polynôme augmente car la taille des carrés croît en conséquence. En effet, les points des carrés contenant des contours (indiqués en hachuré sur la figure) ne peuvent être associés qu'aux régions avoisinantes nées du rassemblement des carrés ne contenant pas de contours ce qui rend impossible l'apparition de zones homogènes provenant uniquement de l'association de points des carrés contenant des contours (les points constituant l'ellipse dans l'exemple de la figure 4.11). Pour éviter cette distorsion, il faudra regrouper les pixels correspondant à ces zones homogènes allongées sur la base d'un certain critère, vu qu'il ne faut pas les inclure directement dans le GCR (voir paragraphe 4.5.3). Pour pouvoir décrire les régions ainsi constituées au moyen du même modèle d'approximation polynomiale, il faut que le nombre de pixels de ces régions soit supérieur au nombre de coefficients du polynôme.

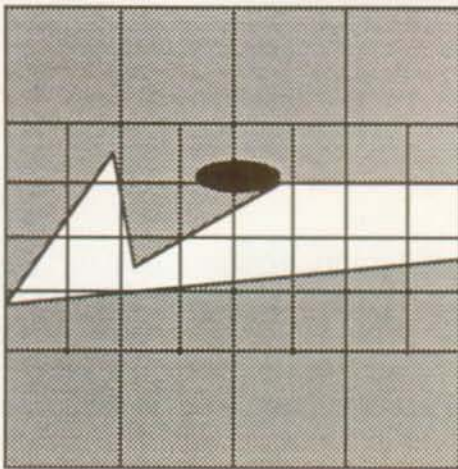
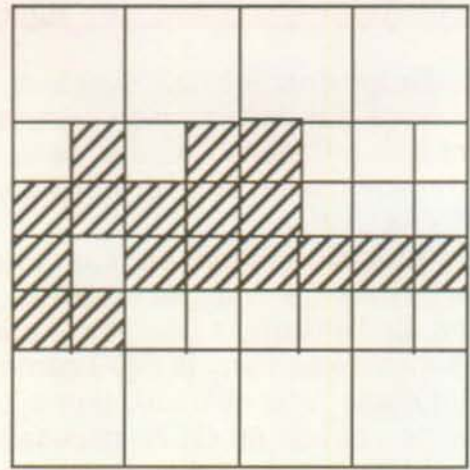
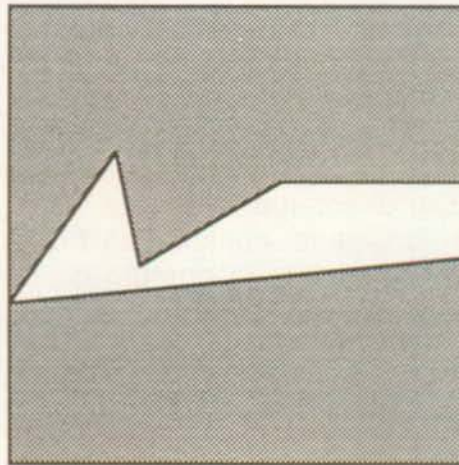


Image originale avec
partition en surimpression
(4 régions)



Carrés contenant des contours
(en hachuré)



Segmentation finale
(3 régions)

Figure 4.11: Disparition des zones homogènes allongées.

En pratique, on risque de voir apparaître un autre genre d'artefact. En effet, prenons le cas d'un carré de taille 4x4 contenant un contour qui a été subdivisé sur la base d'une image de contrôle des structures positives et négatives en 2 régions contenant chacune 10 et 6 points. Chacune de ces petites régions appartient à un espace régulier différent car elles sont séparées par un contour. Il est cependant possible de trouver un polynôme de degré 2 qui suive avec très peu d'erreur le carré qui pourrait être réobtenu par rassemblement de ces deux régions. Si ce carré se trouve ensuite inséré dans une région de plus grande taille, le contour disparaît. On étudiera au paragraphe 4.5.3 comment on peut éliminer un tel défaut en changeant pendant quelques étapes de rassemblement, le degré de l'approximation et en essayant d'associer le plus grand nombre possible de pixels voisins situés d'un même côté d'un contour pour empêcher que ceux-ci ne puissent être associés au cours d'un rassemblement ultérieur aux pixels situés de l'autre côté du contour.

Maintenant que la liste des défauts a été mise en évidence, décrivons la stratégie globale de rassemblement qui permet de s'approcher d'une segmentation idéale, c'est-à-dire qui n'est constituée que de régions représentant des objets de la scène.

4.5.2 Regroupement des régions sans contours :

Il s'agit de la première phase de la stratégie de rassemblement. Partant de la partition donnée par l'algorithme de division, on va unir les carrés voisins ne contenant pas de contours de manière à faire apparaître des zones homogènes de l'image. Le rassemblement de deux régions voisines se fait sur la base du minimum de dissimilarité du graphe de contiguïté de région. Le graphe de contiguïté de région peut être non connexe, vu qu'il peut ne pas exister de chemin entre deux portions du GCR.

Sur la base de la procédure d'expérimentation des critères de dissimilarité, il apparaît intéressant d'utiliser dans un premier temps un critère qui permette d'obtenir des espaces réguliers de grande dimension. Les critères EQP ou EQCP semblent parfaitement convenir à cette tâche. Vu le comportement très proche de ces deux critères, le choix a porté sur EQCP pour des raisons de moins grande complexité algorithmique (voir paragraphe 4.2.3) et du fait que EQCP permet de mesurer à chaque étape la distorsion exacte qui est introduite entre les approximations avant rassemblement et après rassemblement.

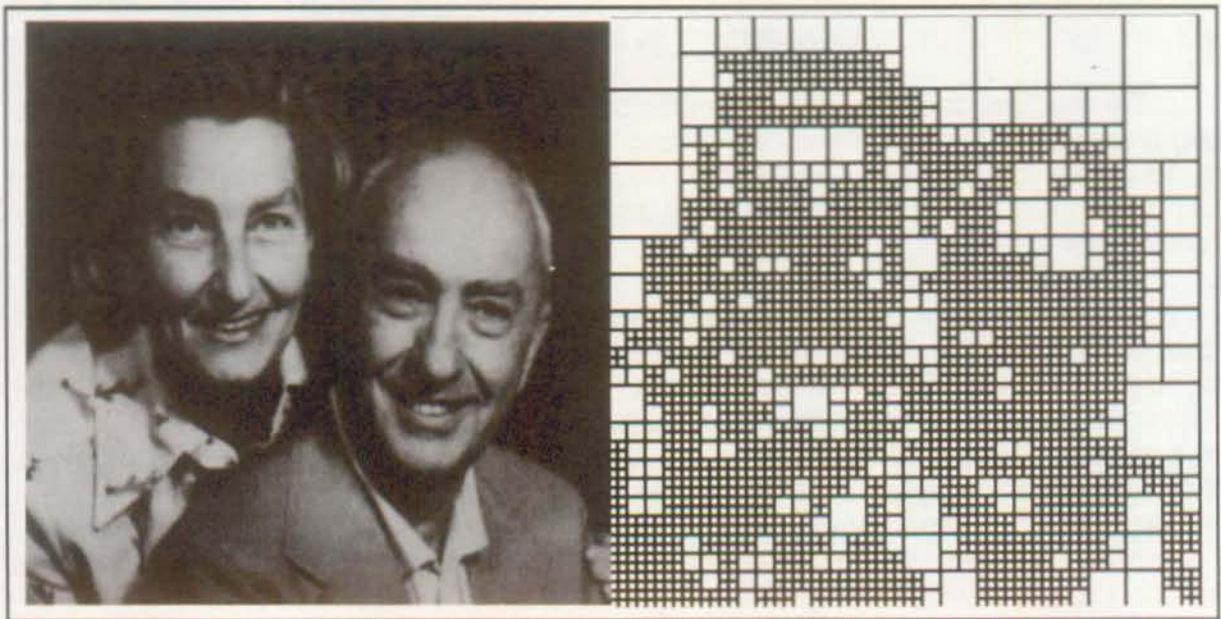
De manière à éviter l'apparition de petites enclaves isolées dans des espaces réguliers de grande dimension, il faut également utiliser une mesure d'erreur quadratique ou d'erreur quadratique courante. De nouveau, et vu les meilleures performances obtenues, on a choisi une mesure d'erreur quadratique courante (voir figure 4.10).

A la fin de cette première phase, l'image est constituée d'une série de régions dont la forme a été choisie de manière à s'adapter au mieux au modèle polynomial de degré choisi. Un point important reste à définir. Combien d'étapes de rassemblement peuvent avoir lieu entre la modification des critères de dissimilarité et quand est-ce qu'il faut arrêter cette première phase de l'algorithme? Il est plus exact de poser cette question en termes du nombre de régions dont est constituée la segmentation de manière à se rendre assez peu dépendant du degré du polynôme d'approximation. En effet, le nombre de régions du GCR initial est une fonction décroissante du degré du polynôme d'approximation.

La réponse à la première question peut être trouvée en analysant les résultats de l'expérience du paragraphe 4.5.1.1. On remarque qu'avec environ 1000 régions, le critère d'erreur quadratique pondérée permet pratiquement d'atteindre le but recherché et que vers 200 régions, on commence à voir apparaître certaines distorsions. Il ne faut pas oublier aussi qu'on n'a pas encore inséré ici au GCR de régions correspondant aux carrés contenant des contours. Ceci justifie de prendre une valeur inférieure à 1000. De manière à tenir compte également de la variabilité du nombre initial de régions de la partition par division de différentes images naturelles, on a choisi de fixer ce seuil à environ 500 régions quelle que soit l'image traitée.

La réponse à la deuxième question dépend du nombre de parties disjointes du GCR. En effet, il est impossible d'arrêter le processus de rassemblement lorsqu'on a obtenu une segmentation formée d'un nombre de régions inférieur au nombre de parties disjointes du GCR. Comme la segmentation que l'on aura après cette première phase de l'algorithme n'est pas celle désirée (vu que les frontières des régions ne correspondent pas à des contours de l'image), il faut qu'elle soit formée d'un nombre suffisamment élevé de régions de manière à ne pas détruire l'optimalité de la procédure adaptative de rassemblement pour la suite de l'algorithme. Ainsi, on a décidé qu'on pouvait passer à une phase ultérieure de l'algorithme lorsque le nombre de régions dont est formée la segmentation de l'image est inférieur à environ 300 régions.

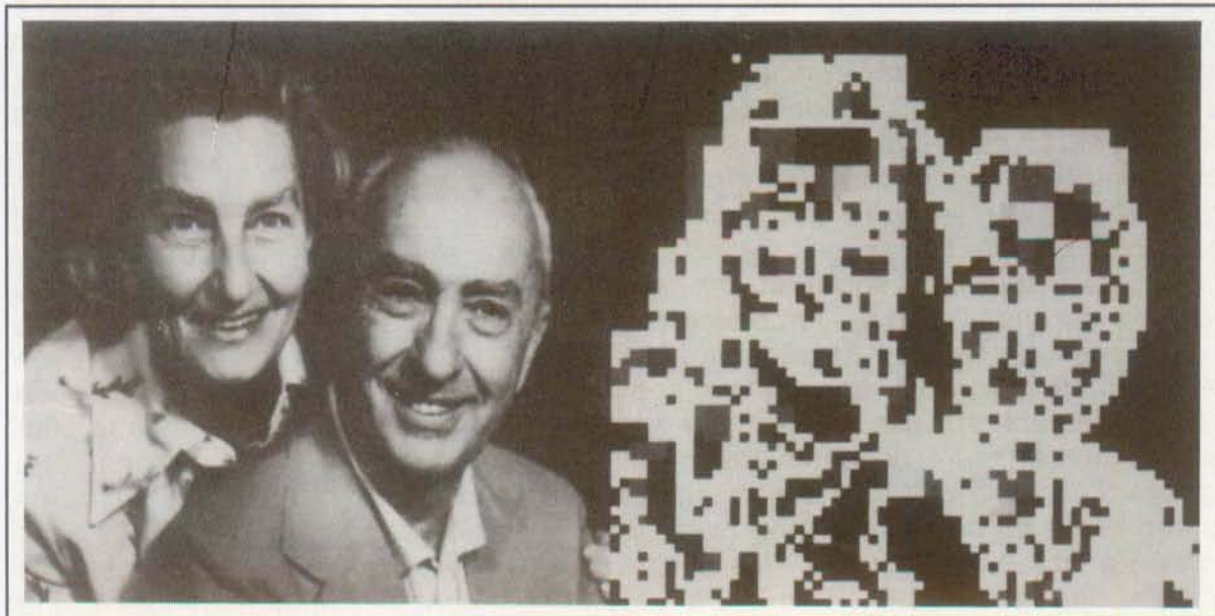
La figure 4.12 présente la forme de la segmentation obtenue pour l'image du couple après cette première phase de l'algorithme. Le polynôme d'approximation est de degré 3. La figure 4.12a montre l'image reconstruite après application de l'algorithme de division et la partition correspondante. La figure 4.12b indique la forme de la segmentation constituée de 499 régions et l'image reconstruite correspondante après rassemblement avec utilisation du critère d'erreur quadratique courante pondérée. La figure 4.12c illustre la segmentation obtenue après application de l'erreur quadratique courante. L'image est alors segmentée en 299 régions plus le nombre de carrés contenant des contours. Les parties correspondant à ces carrés sont indiquées en blanc donnant naissance à des images à 6 niveaux de gris pour décrire la forme de la segmentation.



a) Image approximée après division et partition correspondante



b) Image reconstruite après rassemblement par EQCP et partition correspondante (499 régions ; les carrés contenant des contours sont indiqués en blanc)



c) Image reconstruite après rassemblement par EQC et partition correspondante (299 régions ; les carrés contenant des contours sont indiqués en blanc)

Figure 4.12: Première phase de la segmentation
Regroupement des carrés ne contenant pas des contours.

4.5.3 Insertion dans le GCR des régions adjacentes aux contours :

A partir de la segmentation précédemment obtenue, il s'agit d'associer à partir de maintenant les régions se trouvant au voisinage des contours de l'image aux espaces réguliers qui leur correspondent le plus en utilisant la procédure adaptative de rassemblement. Ces régions sont définies à partir des carrés

contenant un contour par l'ensemble des pixels situés du même côté de ce contour. Pour les définir, on peut avantageusement utiliser l'image des structures positives et négatives de l'image originale. On superpose à cette image l'ensemble des carrés de plus petite taille après division de l'image originale, contenant des contours. Par continuité, on crée alors une région constituée par l'ensemble des points appartenant à des carrés contigus, points qui possèdent le même signe sur l'image des structures positives et négatives. S'il existe une approximation polynomiale de même degré que celle utilisée pour le rassemblement, on insère la région ainsi générée dans le GCR. S'il n'existe pas une seule approximation au sens des moindres carrés (la matrice S associée à cette région est singulière), on inclut un à un les points de cette région comme noeuds du GCR avec la contrainte implicite qu'ils ne peuvent être rassemblés qu'à des régions où l'approximation existe (voir paragraphe 4.1.1). Cette procédure d'insertion est terminée lorsqu'on a analysé tous les points des carrés contenant des contours.

Il est à remarquer que certaines régions ainsi obtenues peuvent correspondre à des zones homogènes à part entière qui ne doivent pas être regroupées aux régions obtenues lors de la première phase du rassemblement (voir distorsions causées par l'augmentation du degré du polynôme du paragraphe 4.5.1.2).

En décidant d'associer le plus grand nombre possible de points situés d'un même côté d'un contour, on rend difficile l'association de la région ainsi constituée à la région existant de l'autre côté du contour. En effet, l'erreur de reconstruction introduite par le rassemblement de ces deux régions est beaucoup trop importante. Il peut cependant exister des configurations où le nombre des points de cette régions après rassemblement reste du même ordre de grandeur que le nombre de coefficients du polynôme d'approximation. Dans de tels cas, l'erreur de reconstitution n'est pas nécessairement élevée, rendant le rassemblement possible. On peut observer les artefacts qui en résultent à la figure 4.13, le polynôme d'approximation étant de degré 3, la segmentation étant formée de 99 régions obtenues par rassemblement adaptatif par EQC (observer les erreurs commises sur l'épaule de l'homme). On peut pallier à cet inconvénient, en réduisant le degré du polynôme d'approximation pendant quelques étapes de rassemblement; en l'occurrence il sera fixé à 0. Les régions où ce phénomène peut se manifester étant de petite taille, elle ne peuvent que très difficilement être rassemblées à des régions correspondant à d'autres espaces réguliers après changement du polynôme d'approximation. L'insertion des N régions créées avec l'aide de l'image de contrôle des structures positives et négatives fait passer le nombre de régions du GCR de 300 à $300+N$. Il est donc raisonnable de tolérer une modification du degré du polynôme d'approximation jusqu'à obtenir une segmentation constituée de 300 régions. De manière à éviter l'existence de petites régions et pour favoriser l'insertion des régions que l'on vient de générer aux 300 régions initialement obtenues le critère de dissimilarité choisi est l'erreur quadratique courante.

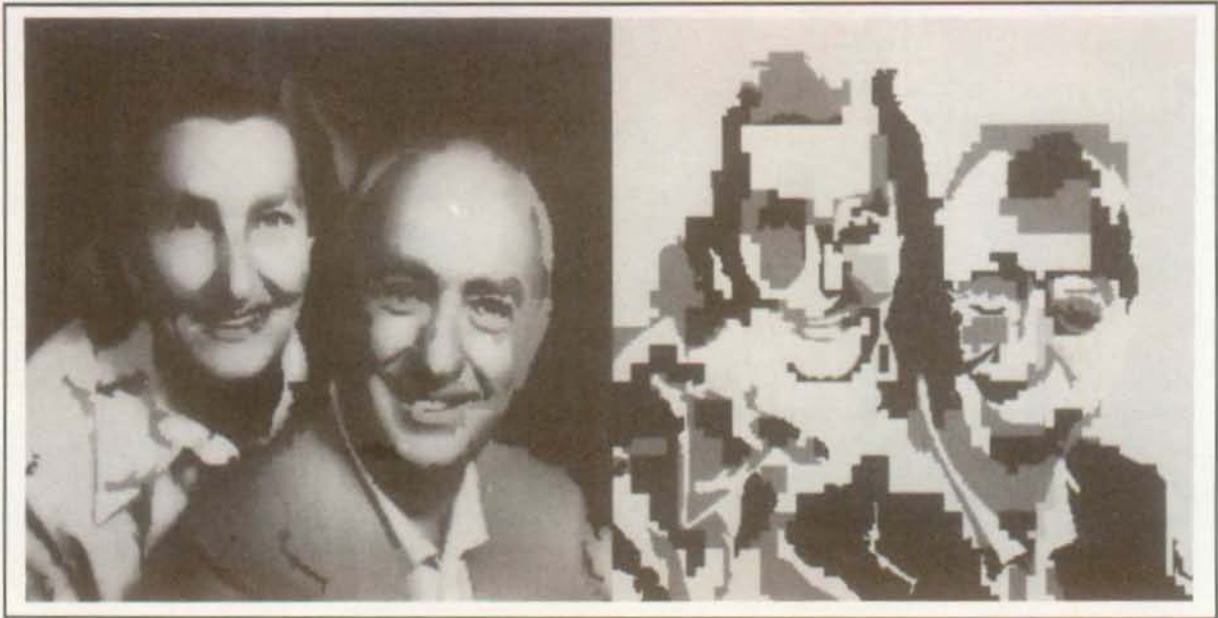
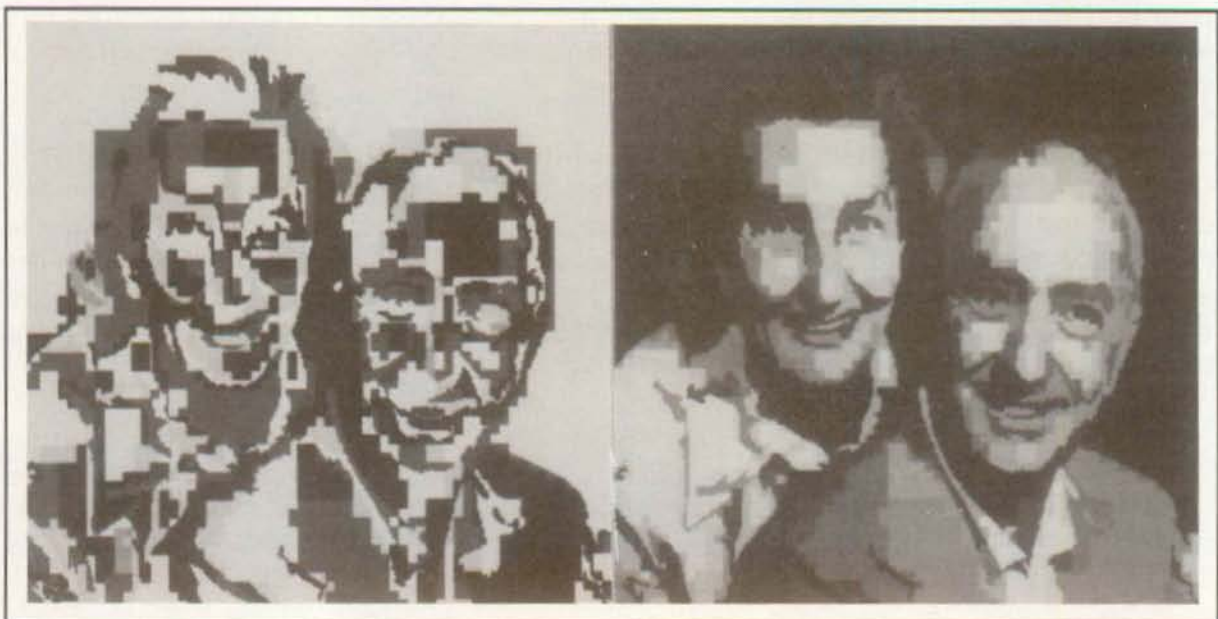


Figure 4.13: Rassemblement par EQC sans modification du degré du polynôme d'approximation de 3 à 0 (segmentation formée de 99 régions). Remarquer les distorsions sur l'épaule gauche de l'homme.

La figure 4.14 illustre les résultats d'une telle procédure sur l'image du couple lorsque l'on a ramené le nombre de noeuds du GCR à 299 régions. On est parti des résultats de la figure 4.12. La figure 4.14a montre la forme de la segmentation, la figure 4.14b montre l'image reconstituée au moyen de polynômes de degré 0 alors qu'à la figure 4.14c, on a l'image reconstruite par des polynômes de degré 3.



a) Forme de la segmentation

b) Image approximée par des constantes.



c) Image reconstruite par des polynômes de degré 3.

Figure 4.14: Rassemblement par EQC avec modification du polynôme d'approximation des régions correspondant aux carrés contenant des contours. Segmentation formée de 299 régions.

4.5.4 Rassemblement final:

En partant d'un GCR à environ 300 régions approximées chacune au moyen d'un polynôme d'un même degré, il s'agit à ce niveau de réduire une dernière fois le nombre de noeuds par rassemblement adaptatif selon un critère d'erreur quadratique courante.

Le critère d'arrêt peut être fixé en liaison avec le niveau de détail souhaité. Si l'on s'intéresse à des applications de codage d'images, le facteur de compression est une fonction décroissante du nombre de régions. On peut ainsi réduire le nombre de régions tant qu'on n'a pas atteint la compression demandée. Pour des applications d'analyse de scène, il ne faut pas trop réduire le nombre de régions de manière à éviter de perdre des détails essentiels à la compréhension de la scène (yeux des personnages dans l'image du couple).

La réduction du nombre de région entraîne nécessairement une réduction de la qualité de l'image. Les figures 4.15, 4.16, 4.17 et 4.18 illustrent les résultats de reconstitution pour l'image du couple segmentée avec 199, 99, 49 et 24 régions, respectivement. Le sous-numéro a représente l'image reconstruite alors que le sous-numéro b représente la forme de la segmentation. La réduction du nombre de régions s'accompagne d'une représentation quasi-idéale de la scène, ayant obtenu une segmentation où les frontières des régions définissent des contours des objets de l'image. Au-delà d'un certain nombre de rassemblement, on ne peut cependant éviter la disparition de quelques objets de la scène.

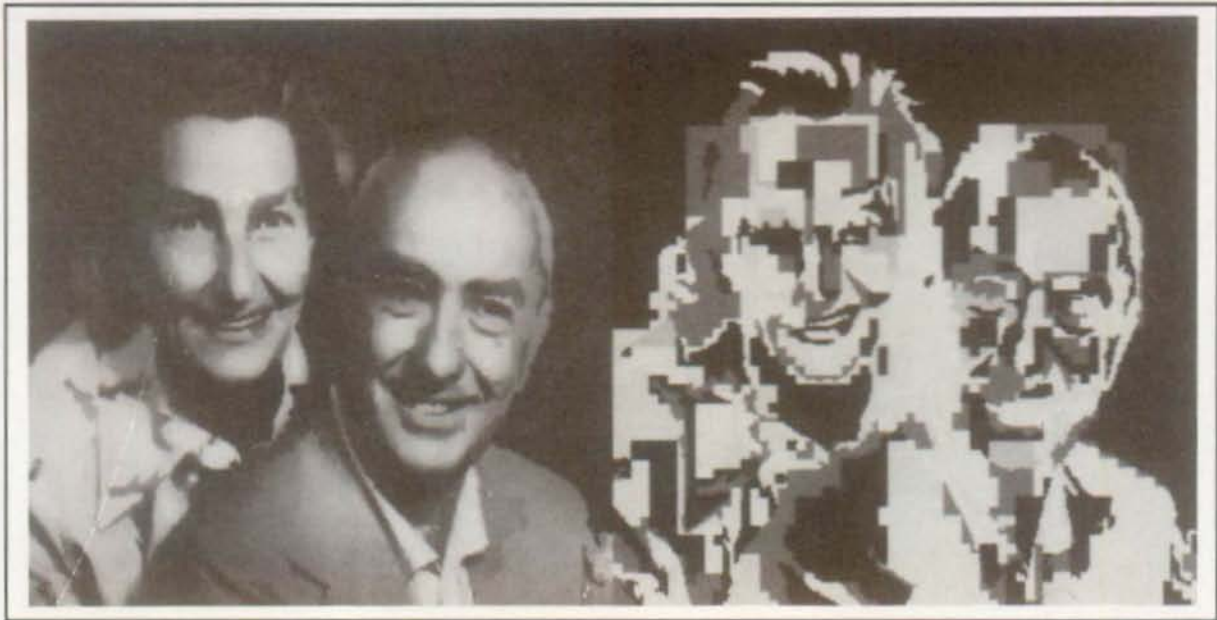


Figure 4.15: Segmentation finale de l'image du couple à 199 régions.

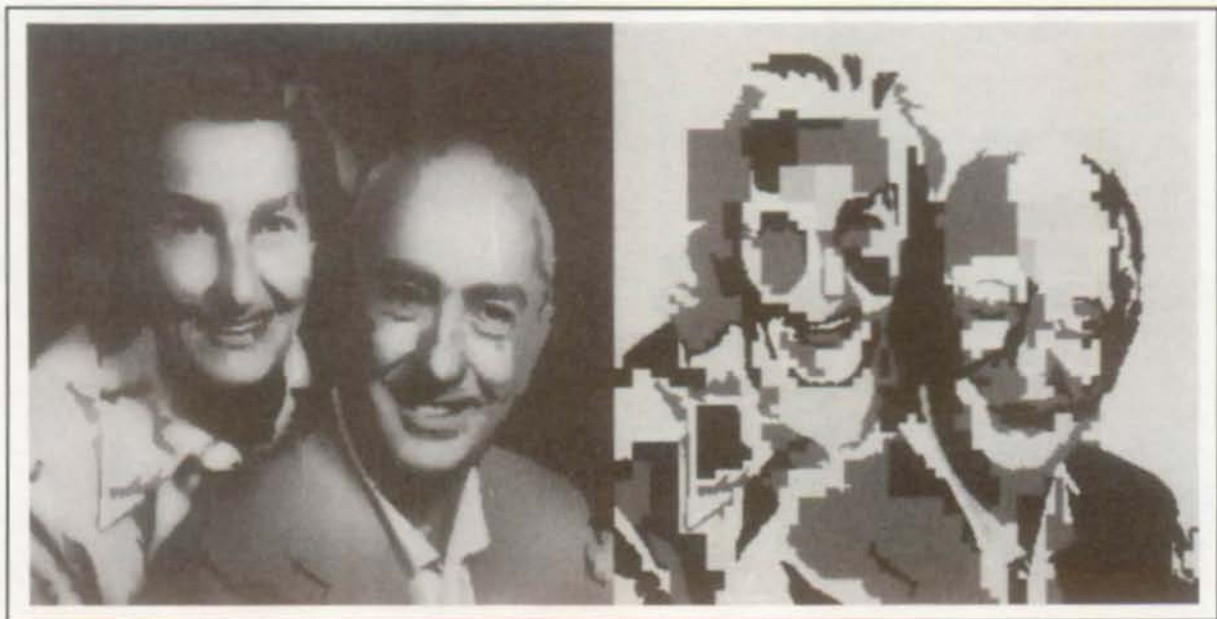


Figure 4.16: Segmentation finale de l'image du couple à 99 régions.

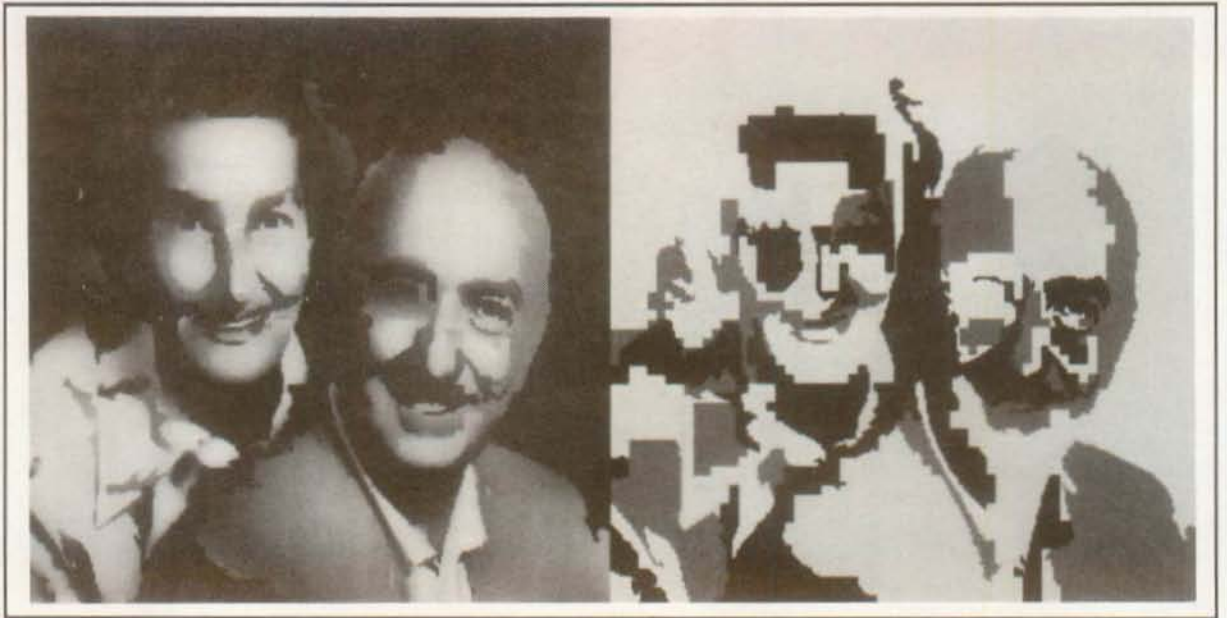


Figure 4.17: Segmentation finale de l'image du couple à 49 régions.



Figure 4.18: Segmentation finale de l'image du couple à 24 régions.

La figure 4.19 résume la stratégie de rassemblement que nous avons définie.

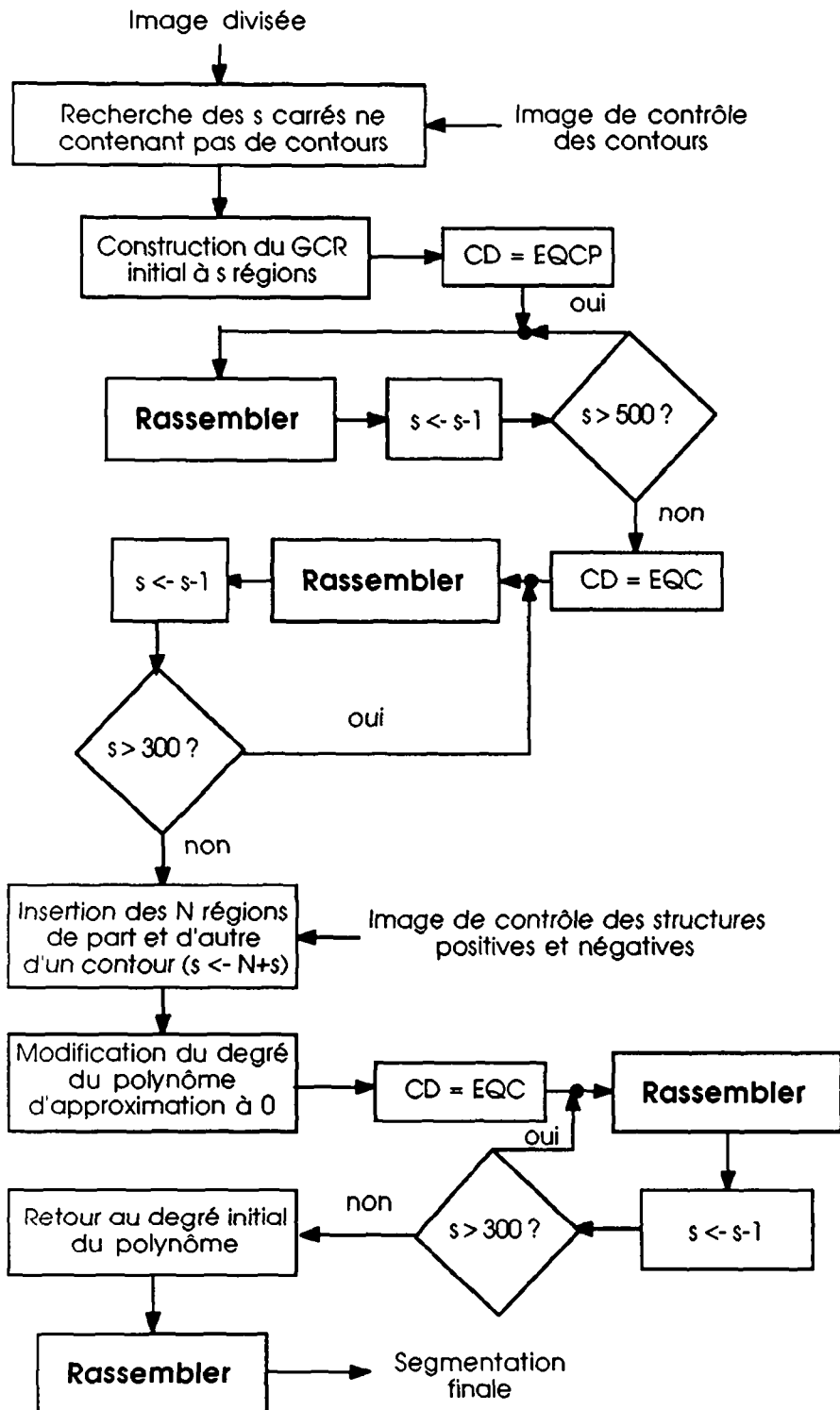


Figure 4.19: Stratégie globale de rassemblement
(CD = critère de dissimilarité ; s = nombre de régions)

4.6 Conclusion :

Ce chapitre a permis de définir la deuxième phase de la segmentation : le rassemblement adaptatif des régions.

Le principe de l'algorithme de rassemblement dont les bases ont été édictées au chapitre 2 consiste à rechercher à chaque étape de rassemblement le minimum de dissimilarité du graphe de contiguïté de région (GCR).

La première section de ce chapitre introduit les contraintes nécessaires à la mise en correspondance des frontières des régions avec les contours réels des objets de l'image. Elle propose ensuite différents critères de dissimilarité pour comparer les régions de l'image. Enfin, elle discute du critère d'arrêt de la procédure de rassemblement.

La section 4.2 met l'accent sur la propriété de décomposition de l'approximation au sens des moindres carrés. Cette propriété permet de calculer efficacement l'approximation du signal original sur une région née du rassemblement de deux régions à partir des approximations calculées sur chaque région impliquée dans le rassemblement. Le calcul des mesures de dissimilarité peut se faire à partir de la connaissance des erreurs calculées sur chaque région prise séparément.

La section 4.3 met l'accent sur la structure des données appelée "brain" permettant l'implantation de l'algorithme alors que la section 4.4 en décrit la complexité et propose des solutions sous-optimales qui accélèrent l'exécution mais ne réduisent pas les performances.

Enfin, la section 4.5 définit la stratégie globale de rassemblement pour obtenir une segmentation optimale. L'idée essentielle consiste à jumeler deux critères de dissimilarité ayant des comportements complémentaires au cours du processus de rassemblement. L'erreur quadratique courante pondérée permet dans un premier temps d'obtenir une partition de l'image avec des espaces réguliers de grande dimension. L'erreur quadratique courante va rattacher les régions de plus petite taille à ces espaces ou entre elles de manière à former les objets de l'image. Cette section indique comment modifier l'algorithme de segmentation pour rendre possible le rassemblement lorsqu'on augmente le degré du polynôme d'approximation.

Maintenant que la segmentation a été obtenue, il s'agit d'étudier comment coder une image représentée par un ensemble de régions approximée par une certaine fonction polynomiale.

5. Représentation de l'information segmentée et postraitement

Une fois segmentée par l'algorithme adaptatif de division et rassemblement, une image est formée par un ensemble de régions adjacentes. Chaque région est limitée par sa frontière constituée d'un ensemble de points 4-connexes définissant l'information de contour. A l'intérieur d'une région, le niveau de gris est représenté par une fonction polynomiale bidimensionnelle approximant au sens des moindres carrés le signal original. Les coefficients polynomiaux correspondent donc à l'information texture du modèle contour-texture. Toute représentation d'une image ainsi obtenue passe donc par la description des frontières de régions et du modèle d'approximation permettant de reconstituer le signal luminance à l'intérieur de chaque région.

La section 5.1 met en évidence comment coder l'information contour. La section suivante aborde le problème de la quantification des coefficients polynomiaux liés à l'information texture. Les deux dernières sections de ce chapitre illustrent les postraitements qui peuvent être effectués pour améliorer la qualité de l'image reconstituée. La section 5.3 montre comment faire disparaître les faux contours.

5.1 Codage de l'information contour :

On appelle *points frontières* d'une région les points qui n'ont pas comme voisins que des points de cette même région. Une description complète de tous les points frontières de chaque région est redondante car la connaissance d'une frontière d'une région définit automatiquement les parties correspondantes des frontières des régions qui lui sont voisines. Ceci est illustré à la figure 5.1.

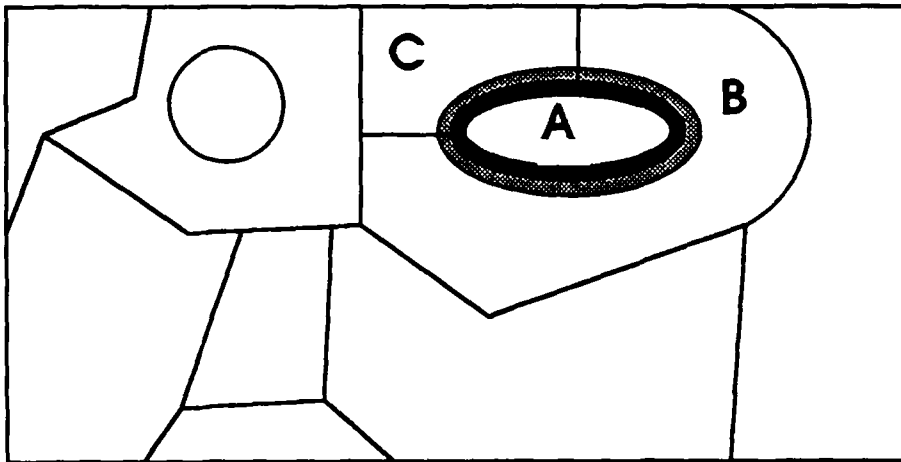


Image segmentée frontière de la région A
 portions des frontières des régions B et C connues après description de la région A.

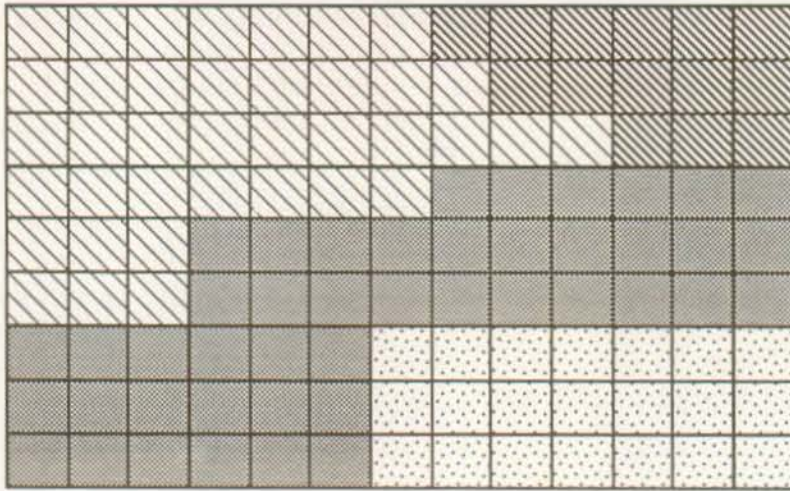
Figure 5.1: Définition simultanée des frontières de régions adjacentes.

En décrivant dans un ordre prédéfini les frontières de chaque région, (en prenant par exemple chaque fois la région située la plus en haut à droite), il est

possible de reconstruire la forme de la segmentation sans autre? Si on n'adopte pas une telle relation d'ordre, il devient alors nécessaire de transmettre de l'information supplémentaire. En effet, appelons *noeuds de frontières* les points frontières possédant plus de deux points frontières voisins. Pour déterminer comment est définie chaque région il devient nécessaire de transmettre pour chaque portion de frontière comprise entre deux noeuds de frontières la région à laquelle cette portion de frontière appartient.

Décider de décrire les frontières de chaque région d'une façon ordonnée, région par région selon un certain ordre de balayage présente cependant toute une série de désavantages:

1.- La forme des régions pouvant être à priori quelconque, il existe des points frontières qui ont plus de deux points frontières voisins appartenant à cette même région. On appellera ces points *points doubles* notés D. De même, il existe des points frontières ne possédant qu'un seul point voisin appartenant à la même région, voir aucun si la région qu'il définit correspond à un pixel isolé. Ces points seront appelés *points isolés* notés I. La reconstitution correcte de la forme de la segmentation (information contour) nécessite de transmettre une frontière appartenant à une certaine région à la fois. Appellons *image binaire des frontières*, l'image des points frontières obtenue après élimination des points frontières des régions adjacentes aux régions prises selon un certain ordre de balayage. Pour éviter toute ambiguïté, il est nécessaire de décrire la forme de l'image binaire des frontières dans l'ordre dans lequel elle a été construite, à cause de la présence de points doubles et des points isolés. Ceci est illustré à la figure 5.2. Une autre possibilité consiste à marquer les points isolés et les points doubles au moyen d'un symbole supplémentaire au niveau du codeur.



Segmentation d'une
image en quatre
régions.

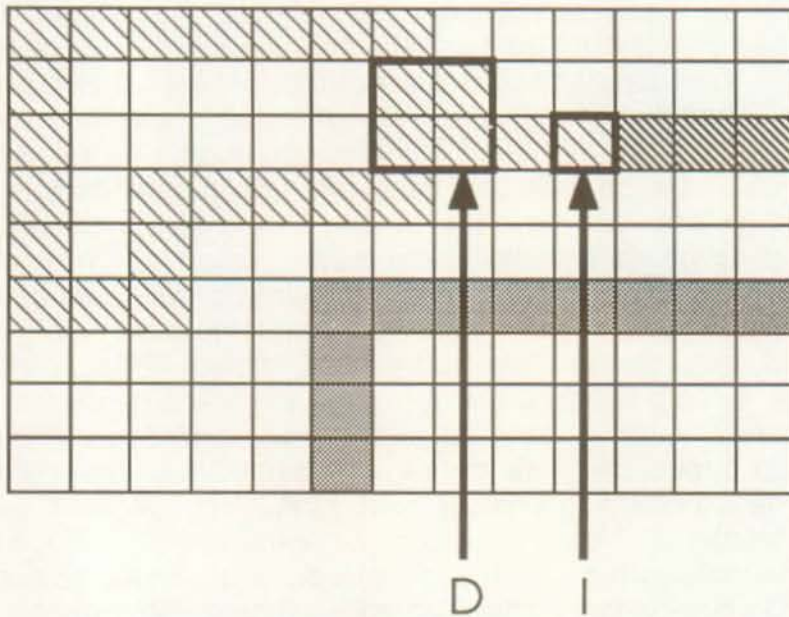


Image binaire des
frontières associée

Figure 5.2: Problèmes causés par la représentation des frontières de régions par une image binaire de frontières. L'image binaire de frontières est calculée en considérant successivement chaque région selon un balayage de gauche à droite puis de haut en bas. Si l'on ne représente que l'information binaire de position des frontières, rien ne permet de distinguer la position du point isolé I. Une situation similaire peut apparaître pour certaines configurations de points doubles, par exemple lorsqu'une portion de frontière relie deux parties d'une même région par un seul chemin, ce chemin apparaissant entre deux points doubles de cette région.

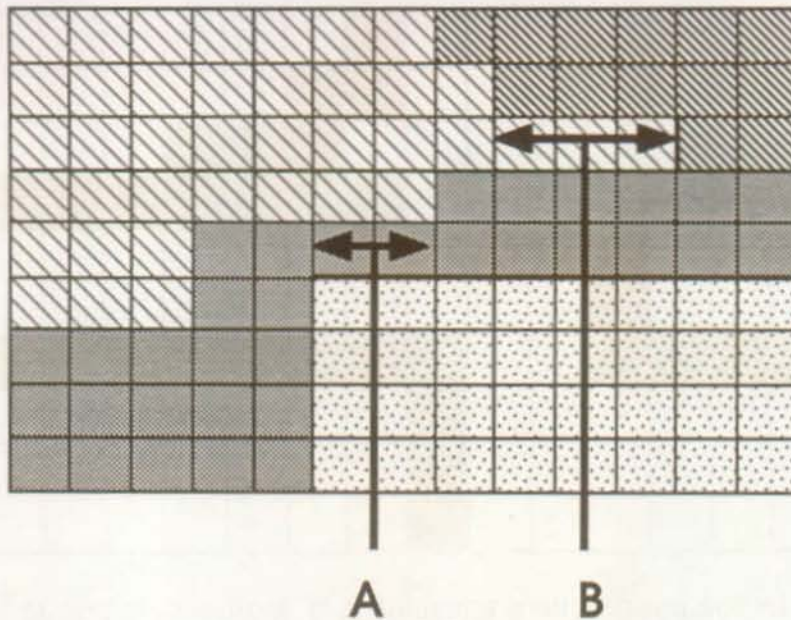
2.- un autre désavantage d'utiliser une approche basée sur une représentation ordonnée des frontières de régions est lié au coût important entraîné par le codage des points doubles ou isolés de chaque frontière de région. En effet, Eden et Kocher ont montré [62] qu'en imposant certaines contraintes géométriques

interdisant l'existence de tels points dans des images binaires de frontières, on peut ramener le coût moyen de 2 bits par point frontière à 1.36 bits par point frontière. Les performances indiquées tiennent compte de la statistique moyenne des symboles à coder et se révèlent être les meilleures par rapport aux autres codes établis pour le codage de l'information contour.

Ces considérations nous ont amené à modifier la forme de la segmentation de manière à obtenir une image binaire des frontières ne contenant pas de points doubles ou isolés. De manière à limiter tant soit peu la modification de la segmentation pour éviter une dégradation inacceptable de l'image reconstruite, un premier pas consiste à remplacer l'image binaire des frontières par une image interfrontière. C'est ce que nous allons décrire dans le paragraphe suivant.

5.1.1 Codage de frontière par image interfrontière:

Au lieu de coder les points frontières de chaque région prise d'une façon ordonnée, il s'agit de décrire la forme des courbes situées entre les points frontières de deux régions adjacentes. A partir de l'image ainsi obtenue appelée image interfrontière, il est parfaitement possible de reconstituer la forme de la segmentation. Les points dont sont constituées ces courbes sont appelés *points interfrontières*. Si l'on apparente ces points interfrontières à des points frontières associés à une segmentation fictive, il n'en est aucun qui définisse un point isolé. Ceci se fait néanmoins au détriment d'une augmentation du nombre de points à coder. En effet, d'une part la taille de l'image interfrontière est de $M+1$ lignes et $N+1$ colonnes si l'image analysée possède M lignes et N colonnes. D'autre part, tout chemin reliant deux points doubles d'une même région ou un point double à un point isolé d'une même région dans l'image binaire de frontières est remplacé par un chemin de largeur double (donc constitué uniquement de points doubles) dans l'image interfrontière, ce qui fait croître d'un facteur 2 le nombre de points correspondant à de tels chemins par rapport à l'image binaire des frontières. De tels chemins seront appelés *liens de régions*. La segmentation de la figure 5.3 montre les deux types de liens de régions possibles. Le coût associé à la représentation de l'information contour sous forme d'image interfrontière ou sous forme d'image binaire de frontières doit rester cependant identique vu qu'aucune modification topologique n'a été encore introduite au niveau de la forme de la segmentation. On a cependant choisi de coder l'image interfrontière parce que la recherche d'un code optimum n'est pas évidente pour une image binaire de frontières sans faire disparaître les points doubles et les points isolés. L'image interfrontière présente l'avantage de ne plus avoir de points doubles à partir du moment où l'on a fait disparaître les liens de régions de l'image segmentée. L'algorithme à utiliser pour y parvenir est décrit au paragraphe 5.1.3.



Segmentation d'une
image en quatre
régions.

Figure 5.3: Forme des liens de régions.

La configuration **A** correspond à un chemin entre deux points doubles d'une même région. La configuration **B** correspond à un chemin entre un point double et un point isolé d'une même région.

L'algorithme utilisé pour générer l'image interfrontière procède de la façon suivante: on balaye dans le sens des coordonnées croissantes l'image de numéros représentant la forme de la segmentation jusqu'à atteindre un point frontière de coordonnée (k,l) . S'il existe un point frontière de coordonnée $(k+1,l)$ appartenant à une autre région que (k,l) , les points de coordonnées $(k,l-1)$ et (k,l) de l'image interfrontière définissent des points interfrontières. De la même façon, s'il existe un point frontière de coordonnées $(k,l+1)$ appartenant à une autre région que (k,l) , les points de coordonnées (k,l) et $(k-1,l)$ définissent à leur tour des points interfrontières.

La figure 5.4 illustre l'image interfrontière associée à la segmentation de la figure 5.2. La figure 5.5 montre l'image interfrontière associée à la segmentation de l'image du couple à 99 régions de la figure 4.16. Remarquer qu'on ne voit pas apparaître de points isolés dans ces figures.

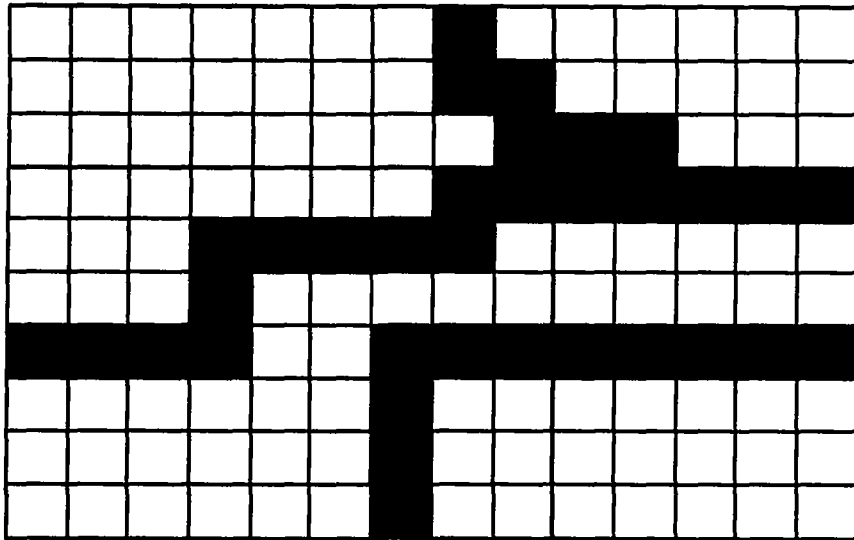


Figure 5.4: Image interfrontière associée à la segmentation de la figure 5.2.

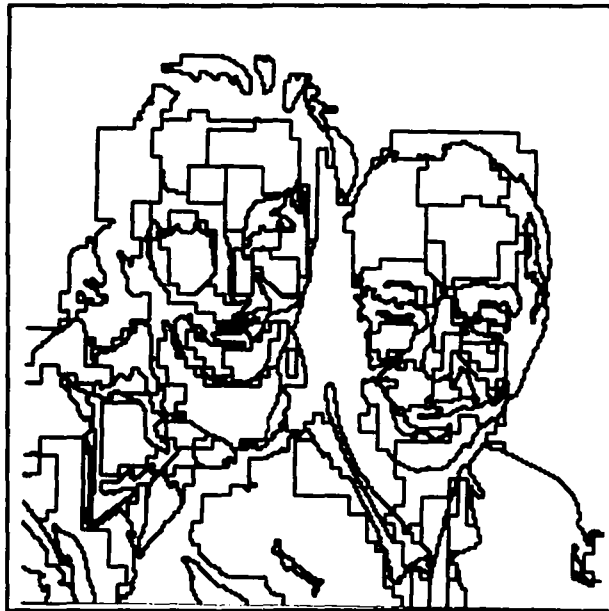


Figure 5.5: Image interfrontière associée à la segmentation du couple de la figure 4.16 (99 régions)

5.1.2 Elimination des liens de régions:

Cette opération va permettre de faire disparaître les points doubles de l'image interfrontière décrivant une certaine segmentation. Cette image constitue dès lors un graphe où les branches représentent les points interfrontières séparant deux régions et les noeuds constituent les points interfrontières séparant plus de deux régions. Vu le contenu sémantique auquel il se rattache, on appellera ce graphe graphe de contours.

Eliminer les liens de régions nécessite une modification de la segmentation. Une question se pose: peut-on tolérer la distorsion de l'image reconstruite

entraînée par une telle modification? Sur le plan visuel, les détails correspondant aux pixels associés à ces liens de régions vont avoir leur valeur modifiée. Or, il nous est apparu à l'observation des images reconstruites que cela n'avait aucun effet réellement perceptible. A cet effet, on présente les images du couple reconstituées pour des segmentations formées de 49 et 99 régions. Ces résultats sont présentés à la figure 5.6. Ils sont à comparer respectivement aux images reconstruites des figures 4.16a et 4.17a.



a) Image reconstruite (99 régions)

b) Image reconstruite (49 régions)

Figure 5.6: Images du couple reconstruites après élimination des liens de régions de la segmentation adaptative par division et rassemblement.

Sur le plan du codage de l'information contour, l'élimination des liens de régions se révèle très efficace pour deux raisons:

- l'élimination des liens de régions entraîne la disparition de tous les points doubles de l'image interfrontière ce qui réduit sensiblement le nombre de points à coder. Ainsi, le nombre de points est passé de $x=8583$ pour l'image interfrontière de la figure 5.5 à $y=7727$ pour l'image interfrontière associée à la figure 5.6a (Voir plus loin, figure 5.9a). Ces images interfrontières sont associées à la segmentation de l'image du couple en 99 régions après application de l'algorithme de division et rassemblement. Dans la seconde, les liens de régions ont été éliminés.

- comme il est montré au paragraphe suivant, le graphe constitué par la nouvelle image interfrontière après élimination des liens de régions peut être efficacement codé, ayant simplifié le choix des symboles nécessaires à ce codage.

Ce gain en compression se fait cependant au détriment de l'information texture car il devient nécessaire d'adresser chaque partie connexe de l'image représentant la forme de la segmentation sans liens de régions à un vecteur de coefficients polynomiaux. En effet, après disparition des liens de régions reliant deux parties d'une même région A de la segmentation après l'application de l'algorithme adaptatif de division et rassemblement, ces deux parties vont être

séparées. Elles sont cependant caractérisées par la même approximation polynomiale définie initialement sur toute la région A. Une étude effectuée au paragraphe suivant montrera que le coût nécessaire à cet adressage reste négligeable par rapport à celui qui serait entraîné par le codage de l'information contour d'une image interfrontière présentant des points doubles.

L'algorithme de modification de la segmentation permettant de faire disparaître les liens de régions fonctionne de la manière suivante: En chaque point de l'image des numéros représentant la forme de la segmentation, on teste si ses deux voisins placés sur la même ligne ou la même colonne portent un autre numéro que lui. Si tel n'est pas le cas, on reproduit ce point avec la même valeur dans une nouvelle image de numéros. Dans le cas contraire, la valeur du point en question est modifiée de manière à prendre systématiquement la valeur du point placé au-dessus de lui s'il possède des voisins différents de lui sur la même colonne que lui. Respectivement, le nouveau numéro est celui du point placé sur sa gauche si les différences apparaissent au niveau d'une ligne. On applique alors la même procédure à la nouvelle image de numéros. La procédure est ainsi itérée tant qu'il existe des différences entre deux images de numéros successives. La figure 5.7 illustre le fonctionnement de cet algorithme.

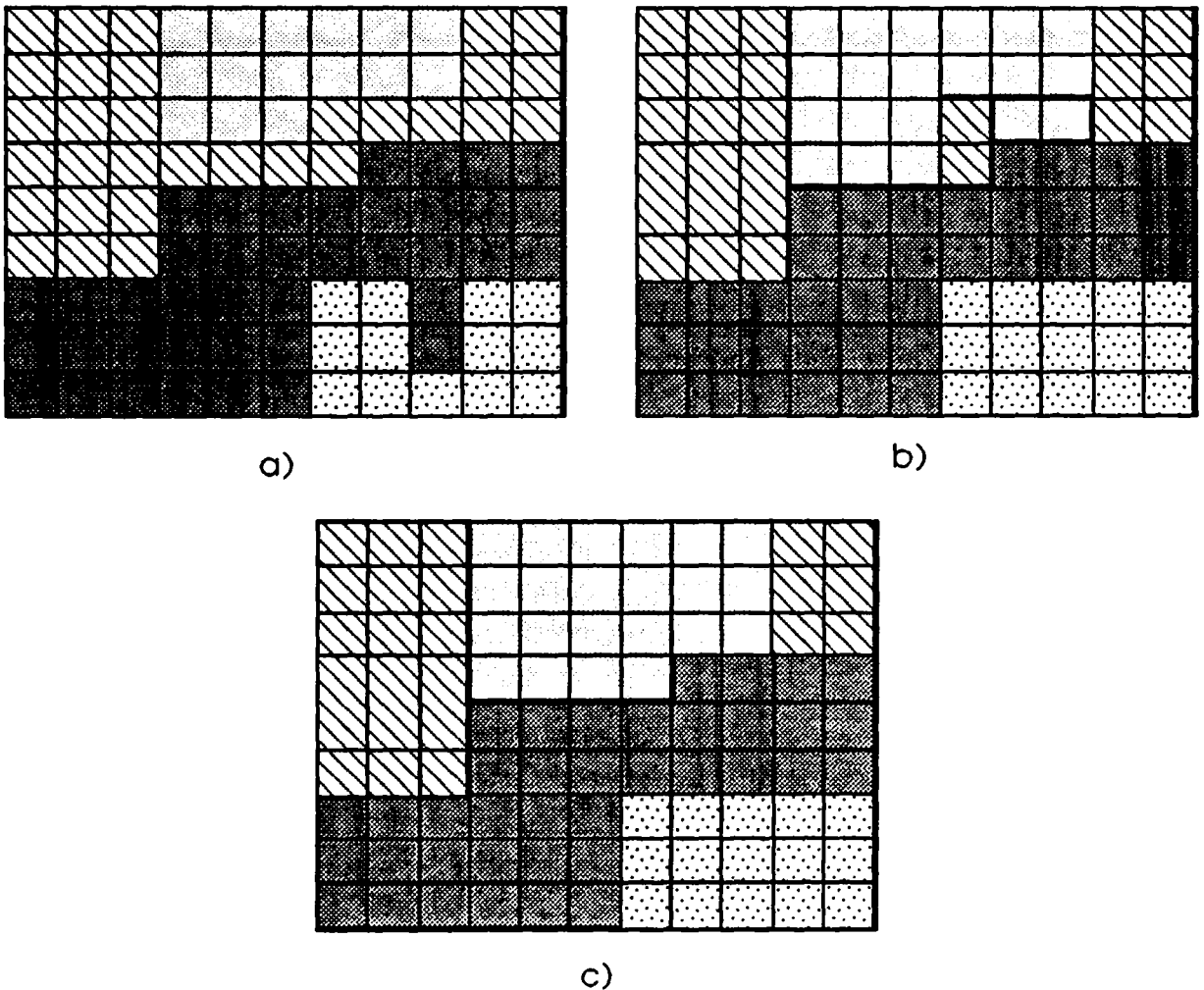
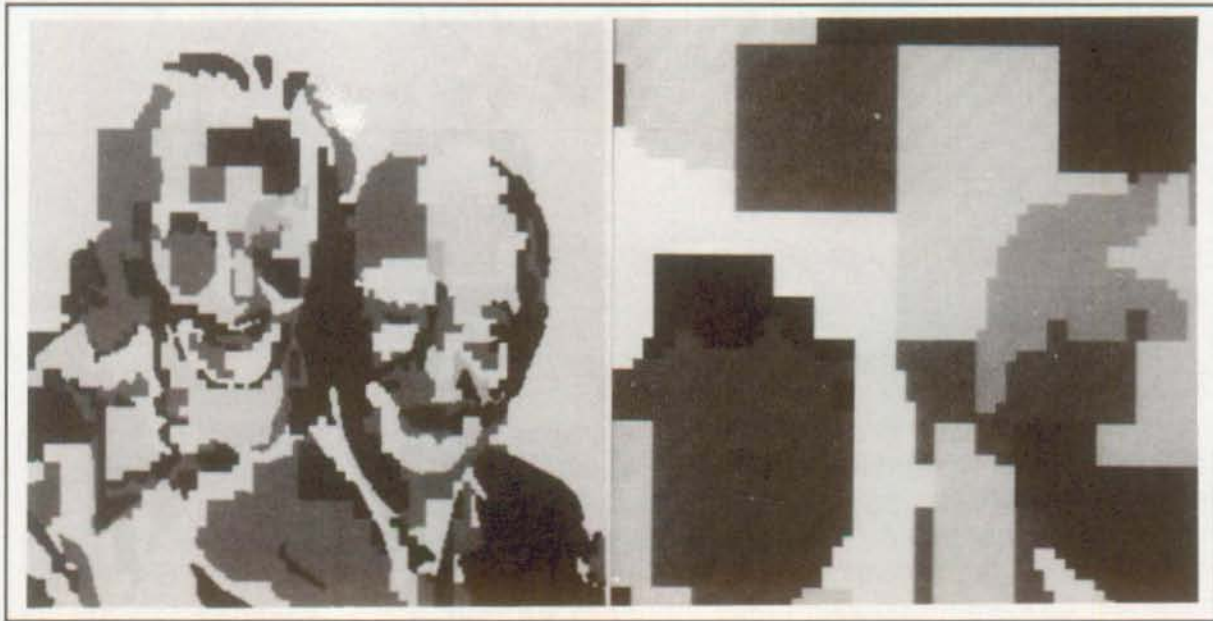


Figure 5.7: Application de l'algorithme d'élimination des liens de régions.

- a) Segmentation initiale
- b) Image des numéros après une itération
- c) Image des numéros après deux itérations

A la première itération, on élimine tout tronçon vertical ou horizontal d'un lien de région, mis à part les points situés à la jonction d'un tronçon vertical et d'un tronçon horizontal. La figure 5.7a présente la forme de la segmentation avant application de l'algorithme. La figure 5.7b présente la forme de la segmentation après une itération. Enfin, la figure 5.7c présente la forme de la segmentation après deux itérations.

La figure 5.8 présente la forme de la segmentation de l'image du couple constituée de 99 régions après élimination des liens de régions. Celle-ci est décrite au moyen de l'algorithme de coloration de chaque région avec au plus cinq niveaux de gris (voir section 4.5). La figure 5.8b montre le détail de la modification entraînée au niveau des yeux de la femme.

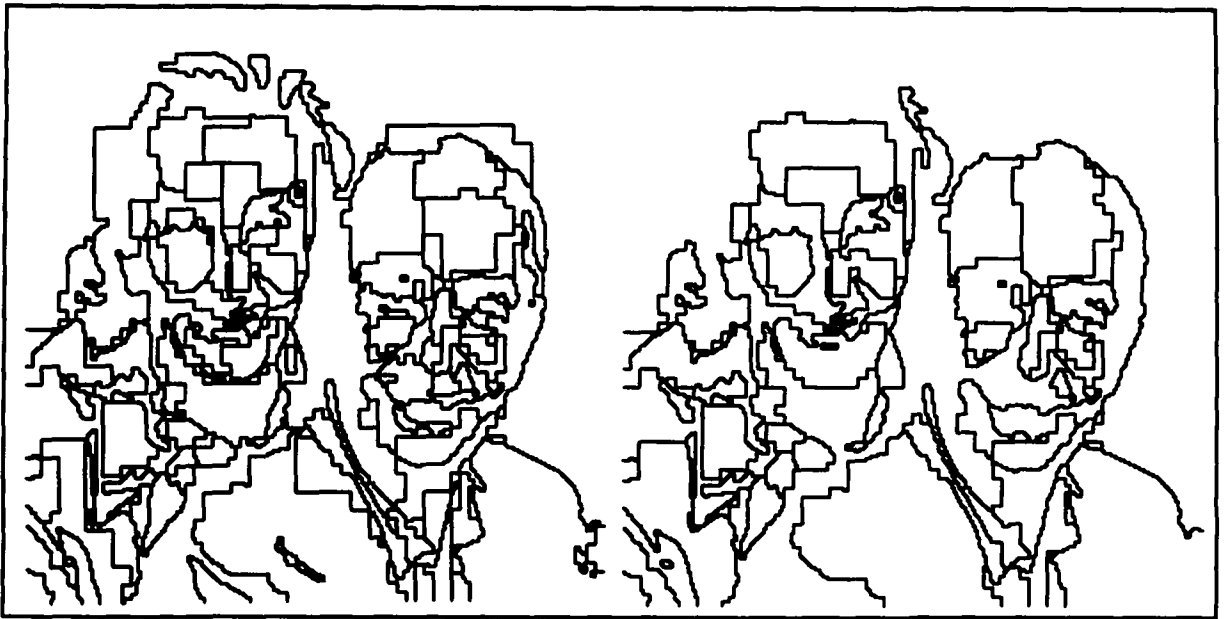


a) Forme de la partition modifiée

b) Détail de l'oeil de la femme

Figure 5.8: Forme de la segmentation du couple à 99 régions après élimination des liens de régions.

Les figures 5.9a et 5.9b montrent les images interfrontières correspondant aux images des figures 5.6a et 5.6b, respectivement (image du couple segmentée à 99 et 49 régions avant élimination des liens de régions).



a) Segmentation à 99 régions

b) Segmentation à 49 régions.

Figure 5.9: Images interfrontières décrivant les segmentations du couple modifiées.

5.1.3 Algorithme de codage du graphe de contours:

Avant de définir comment coder le graphe de contours correspondant à la nouvelle image interfrontière, faisons un léger historique des différentes techniques de codage utilisées pour représenter de tels graphes [63]. Sans parler des méthodes d'extraction de paramètres qui permettent de caractériser grossièrement un contour, les techniques basées sur l'analyse des points contours se distinguent par le fait qu'elles introduisent plus ou moins d'erreur. Vu les exigences imposées au niveau de la correspondance entre les contours réels de l'image et ceux de l'image interfrontière, on ne saurait admettre de codes introduisant des distorsions au graphe de contours. Les travaux de Kocher [27] ont montré à cet effet la sensibilité de l'oeil humain aux erreurs de reconstruction si un contour est décrit par des courbes géométriques simples adjacentes.

Notre choix a dès lors porté sur des techniques de codage sans distorsion aucune, ce qui explique le coût élevé associé à l'information contour. Cette contrainte impose des limitations considérables quant au choix des codes possibles. En remarquant que l'on se déplace selon une certaine direction le long d'un contour, il existe à partir d'un certain point contour connu 8 configurations différentes pour coder la position des points contours voisins non encore connus. Ces configurations sont illustrées à la figure 5.10.

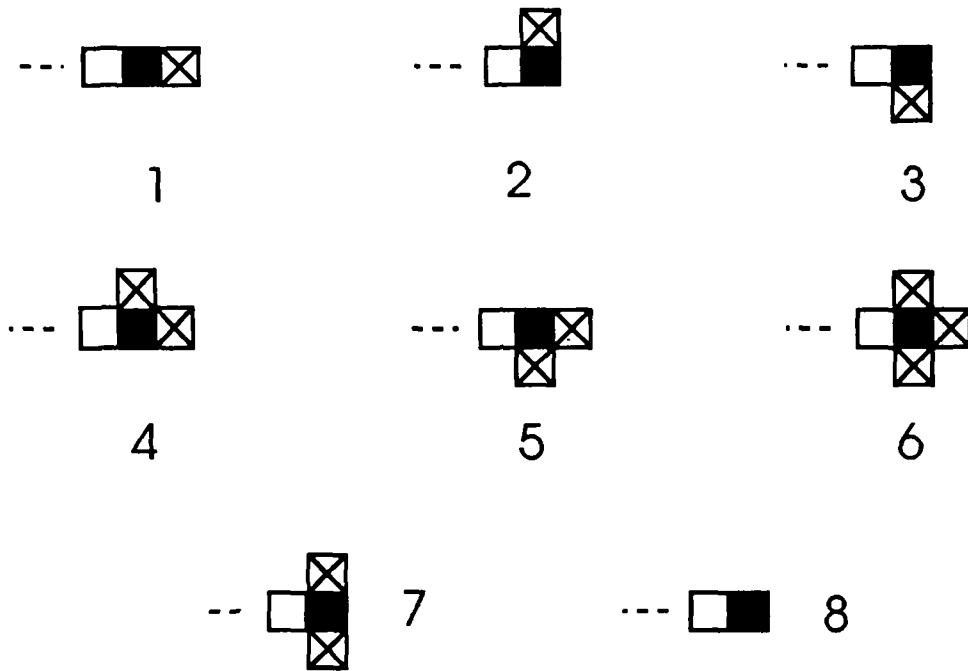


Figure 5.10: Symboles apparaissant dans un graphe de contours 4-connexe.

(le carré noir indique le point couramment traité, le carré blanc le point précédemment traité et le carré marqué d'une croix correspond à la configuration suivante).

Il est possible de coder n'importe quel chemin constitué d'une succession de symboles 1, 2 et 3 avec au plus $\log_2(3) \approx 1.58$ bits par point contour si les symboles 1, 2 et 3 peuvent être admis comme équiprobables. Eden et Kocher [64] ont montré qu'en interdisant l'existence de points doubles le long d'un tel chemin, il ne peut exister de description où les symboles 2 et 3 se suivent, ce qui est équivalent à dire qu'on ne peut effectuer de rotation deux fois de suite dans le même sens. Cette contrainte imposée, on peut modifier après chaque mot code transmis la table d'encodage de manière à ramener le coût d'un point contour à $\log_2(1+\sqrt{2}) \approx 1.27$ bits, toujours dans l'hypothèse d'une équiprobabilité d'apparition de chaque symbole, ce qui s'avère être vrai pour la plupart des contours d'images naturelles à cause des différentes orientations possibles de ceux-ci. Des codes similaires donnent des coûts comparables [65]. A première vue, on peut dès lors décrire un graphe de contour en indiquant au moyen d'une adresse les points correspondant aux symboles 4 à 8 et en utilisant le code précédent pour représenter les chemins constitués par des points contours correspondant aux symboles 1 à 3. En réalité, on peut imposer d'autres contraintes qui permettent d'éviter la plupart de ses adresses. Il suffit pour cela de penser au théorème d'Euler de la théorie des graphes. Ce théorème garantit qu'il est toujours possible de parcourir toutes les branches d'un graphe sans jamais repasser deux fois par la même branche si un nombre pair de branches aboutit à chaque noeud. Ainsi, tout sous-graphe du graphe de contour constitué par les 1 à 3 et 6 peut être décrit en utilisant le même code que précédemment car il n'est pas alors nécessaire de transmettre le symbole 6. Plus généralement, Eden et Kocher [62] ont étudié toutes les contraintes imposées à un graphe de contours sans points doubles. Ils ont montré qu'on pouvait représenter un tel graphe avec environ 1.36 bits par point contour, sans devoir adresser les symboles correspondant aux symboles 4 à 7. Il est seulement obligatoire d'adresser un point pour chaque partie connexe du graphe. En

modifiant après chaque symbole transmis la table d'encodage pour le symbole 1 à 7 à transmettre, on tient compte des densités de probabilités conjointes de ces symboles.

Nous allons montrer qu'en imposant la contrainte de ne pas avoir de points doubles dans un graphe de contours, le gain obtenu est beaucoup plus important par rapport au coût entraîné au niveau des régions devenues disjointes après élimination des liens de régions. En effet, au paragraphe précédent, on a mentionné qu'il devient nécessaire d'indiquer par une adresse quel est le vecteur de coefficient associé à chaque partie de région. L'image interfrontière de la figure 5.5 comprend $x=8583$ points frontières, ce qui peut être codé avec environ $2x=17166$ bits (on a dans ce cas des points doubles). Par contre, l'image interfrontière correspondant à la figure 5.9a comprend $y=7727$ points frontières qui peuvent être codés en utilisant $1.36y=10509$ bits. Le nombre de parties disjointes du graphe est passé de 99 à 138. Il suffit donc d'adresser par au plus un mot de 7 bits chaque région pour savoir à quel vecteur de coefficient elle se rapporte. Le coût supplémentaire associé à l'information texture est donc au plus de $7 \times 138 = 966$ bits ce qui reste beaucoup plus faible que le gain que l'on a obtenu au niveau de l'information contour (6657 bits) par l'élimination des liens de régions.

On a décidé de reprendre le principe du code d'Eden et Kocher si ce n'est qu'il nous a paru plus facile de calculer la table de correspondance entre symboles et mots codes binaires en transformant la connexité du graphe de contours. En effet, les symboles 1 à 3 sont les plus fréquents dans un graphe 4-connexe. La dépendance qui existe entre ces symboles peut être éliminée en remarquant que les configurations 1-1-1, 2-3-2 ou 3-2-3 sont remplacées par un seul et même symbole dans un graphe 8-connexe (tout droit, cf. figure 5.11). De ce fait, le récepteur n'a pas besoin de transformer sa table de correspondance en fonction du symbole qu'il vient de recevoir. Les symboles permettant de suivre un contour sont présentés à la figure 5.11.

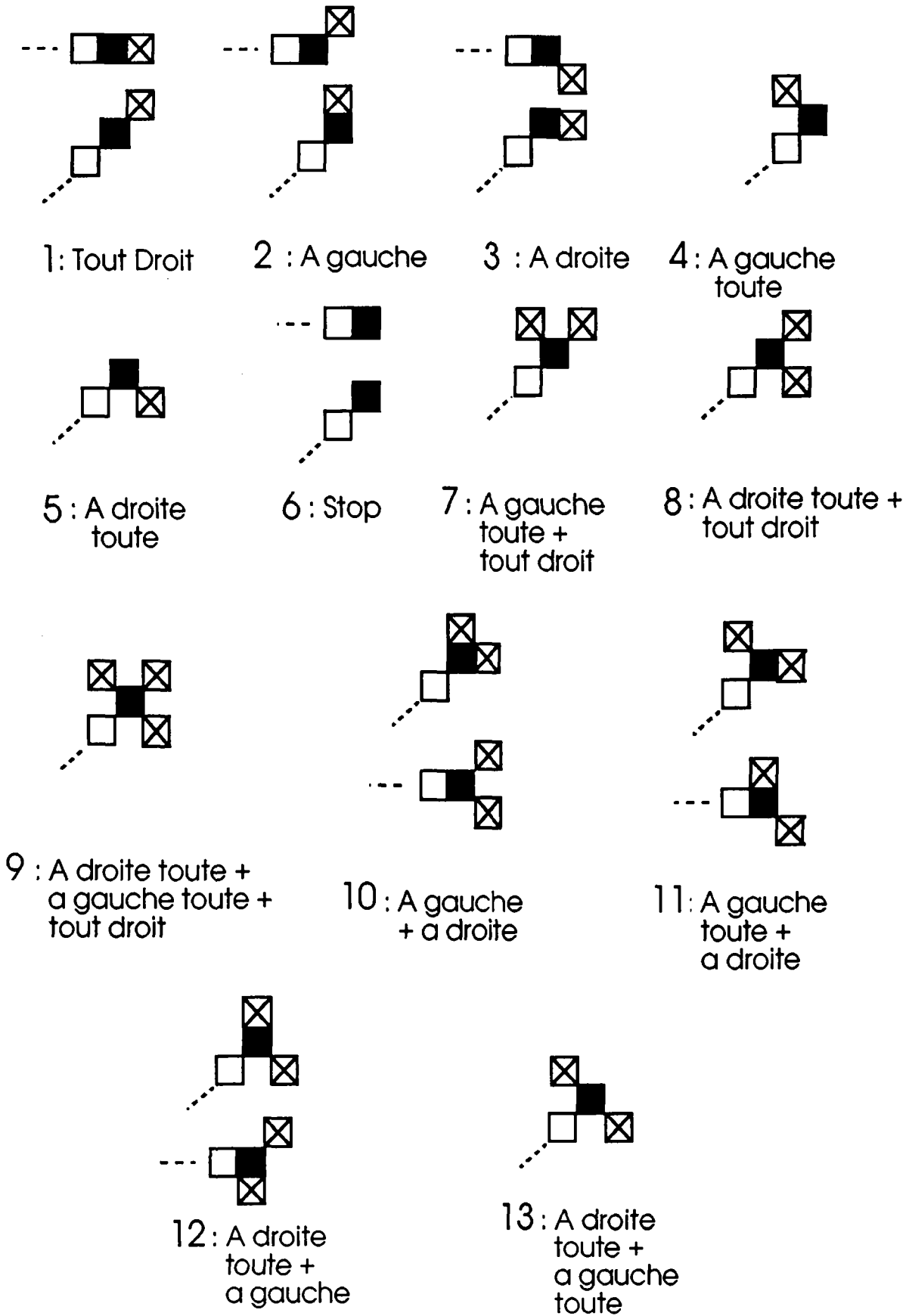


Figure 5.11: Symboles utilisés pour représenter tout graphe de contours 8-connexe (le carré noir indique le point couramment traité, le carré blanc le point traité juste avant et le carré marqué d'une croix correspond à la configuration suivante).

Ces symboles sont utilisés pour représenter tout graphe de contours 8-connexe décrivant la forme de la segmentation sans liens de régions. Pour un graphe de contour donné, chaque symbole est codé selon le code optimal de Huffman. Il est raisonnable d'admettre que le code ainsi obtenu reste optimal pour plusieurs images naturelles car la répartition des probabilités d'apparition des symboles les plus fréquents ne change pas beaucoup d'un graphe de contours à l'autre.

En prenant la segmentation de la figure 5.8a correspondant à une partition de l'image du couple en 99 régions dont on a éliminé les liens de régions, la statistique des symboles obtenue à partir du graphe de contours 8-connexe est donné par les probabilités p_1 à p_{13} suivantes:

$p_1 = 0.5387$	$p_2 = 0.1817$	$p_3 = 0.1541$	$p_4 = 0.0111$
$p_5 = 0.0053$	$p_6 = 0.0535$	$p_7 = 0.0053$	$p_8 = 0.0049$
$p_9 = 0.0002$	$p_{10} = 0.0165$	$p_{11} = 0.0144$	$p_{12} = 0.0112$
$p_{13} = 0.032$			

A cette statistique, on peut associer par codage optimal de Huffman la table de correspondance donnée par l'arbre binaire de la figure 5.12.

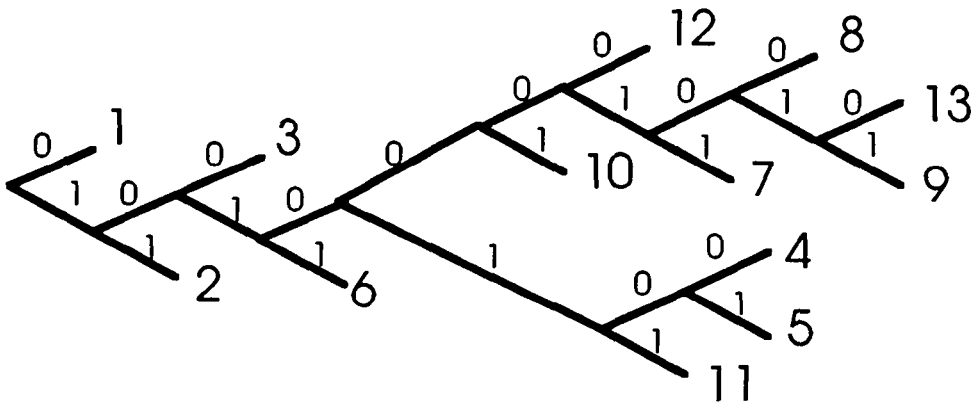


Figure 5.12: Arbre binaire obtenu par codage optimal de Huffman décrivant le mot code associé à chaque symbole.

Il est intéressant de remarquer que la liste des symboles à transmettre présente plusieurs fois consécutivement le symbole "1". Si la longueur moyenne des plages de ce symbole dépasse 2, il est possible qu'il soit plus avantageux de transmettre au moyen de la table de correspondance précédente seulement le premier symbole "1" de chaque plage de "1" (en effet, la table de la figure 5.12 reste optimale car la probabilité d'apparition d'une plage de "1" est donnée par $p_1' = 0.2762$). On utilise ensuite un autre mot binaire pour indiquer la longueur de cette plage. Les performances que l'on peut obtenir avec le code de plages peuvent être grossièrement calculées avec la relation [66]:

$$b \cong r / \log_2 r \quad (5.1)$$

où r indique la longueur moyenne des plages. En réalité, nous utiliserons le code optimal de Huffman pour coder la longueur de chaque plage en étudiant la répartition statistique des longueurs de chaque plage.

Le codage de la segmentation de la figure 5.8a requiert sans utilisation d'un code par plage de symboles "1" 11952 bits et avec utilisation du code de plage de "1" 11869 bits, valeurs auxquelles il faudrait ajouter les $9 \times 16 = 144$ bits pour adresser un point de chaque partie connexe du graphe de contours. Ces valeurs sont supérieures aux estimations que l'on avait faites en utilisant les 1.36 bits par point contour du code de Eden et Kocher. Ceci est essentiellement dû aux hypothèses non parfaitement satisfaites dans ce cas d'image de la répartition statistique de chaque symbole qui ont permis d'arriver à cette estimation.

D'une façon générale, il faut procéder comme suit pour estimer le coût lié au codage de l'information contour.

- compter le nombre de parties connexes du graphe de contours et associer 16 bits pour indiquer l'adresse d'un point de la première partie et autant de bits que nécessaire (moins de 16 bits) pour indiquer la position relative d'un point des autres parties du graphe de contours.

- trouver le code optimal de Huffman pour décrire chaque partie au moyen des symboles 1 à 13. Transmettre 1 bit par partie connexe du graphe pour décrire si l'on utilise un code de plage des symboles "1" ou pas.

- calculer à partir du code de Huffman ainsi obtenu et de la longueur moyenne des plages de "1" (si nécessaire), le coût associé à la description de chaque partie connexe du graphe.

Maintenant que la stratégie de codage a été définie pour décrire l'information contour, nous abordons l'approche nécessaire au codage de l'information texture, c'est-à-dire la quantification des coefficients polynomiaux associés à chaque région.

5.2 Codage de l'information texture :

Ce problème a déjà été mentionné à la section 3.4 lorsqu'il s'agissait de proposer une quantification des coefficients associés à chaque carré de l'image divisée.

Trois étapes vont permettre la caractérisation de l'information texture. La première consiste à montrer qu'en transformant un signal dans un espace de dimension réduite d , il est possible de quantifier chaque vecteur de cet espace avec au plus bd bits où b correspond au nombre de bits avec lequel a été quantifié le signal de départ. Une image segmentée étant constituée d'une série de régions, on étudie ensuite comment réduire les configurations possibles de coefficients en établissant une quantification vectorielle des vecteurs de ces coefficients. Enfin la troisième phase consiste à classer ultérieurement les configurations des vecteurs de coefficients selon 4 classes. Chacune de ces classes se rapportent à un degré de polynôme possible. En effectuant une telle classification, la dimension de l'espace transformé peut être réduite ce qui diminuera en conséquence le nombre de bits nécessaires à la caractérisation de chaque vecteur de coefficients.

Les paragraphes qui suivent décrivent chacune de ces étapes pour aboutir à une méthode de codage.

5.2.1 Coût maximal des coefficients d'approximation:

Considérons un signal numérique g de dimension N qui est transformé dans un espace de dimension r avec $r \leq N$ engendré par r fonctions de base Ψ_j linéairement indépendantes. Le signal transformé u permet de générer une approximation g^\wedge du signal g en utilisant les r fonctions de base Ψ_j . u est calculé de manière à minimiser une certaine fonction d'erreur A entre le signal g et son approximation g^\wedge . En l'occurrence, nous avons choisi le critère d'erreur quadratique (voir section 2.3). Cependant quel que soit le critère, l'approximation g^\wedge se définit toujours par la relation:

$$g^\wedge = Zu \quad (5.2)$$

où la matrice Z caractérise les $N \times r$ valeurs prises par les r fonctions de base Ψ_j aux N points pour lesquels le signal g a été défini. Considérons r échantillons du signal d'approximation défini dans le même espace que g et définissons un nouveau signal de dimension r $g' = g^\wedge$ défini pour les mêmes lieux d'échantillonnage. La relation (5.2) restant toujours valable, on peut trouver sans autre le vecteur transformé u par interpolation des valeurs de g' avec l'ensemble de r fonctions de base Ψ_j en inversant la matrice $r \times r$ Z' associée. Cette matrice est simplement obtenue de la précédente en prenant les r lignes correspondant aux lieux d'échantillonnage choisis pour définir g' . Il est donc possible de *retrouver les $N-r$ valeurs de l'approximation si r de ces valeurs sont connues.*

Si dans l'espace du signal g , les échantillons ont été quantifiés avec b bits, il est raisonnable d'admettre que l'approximation g^\wedge sera quantifiée avec le même nombre de bits. On peut éviter toute erreur de reconstitution en choisissant r valeurs du signal approximé g^\wedge pour lesquelles l'erreur de quantification est nulle. Si cela

n'est pas possible, alors il faut étudier l'expression de l'erreur de quantification aux $N-r$ valeurs du signal approximé reconstitué. Appelons e_q le signal erreur de quantification commis sur les r valeurs du signal approximé g^\wedge constituant le signal g' . Le signal g_q' servant au calcul du vecteur transformé u s'exprime par:

$$g_q' = g' + e_q \quad (5.3)$$

Le vecteur transformé prend alors pour expression:

$$u = Z'^{-1}(g' + e_q) \quad (5.4)$$

ce qui entraîne une erreur de reconstitution au niveau du signal g^\wedge donnée par :

$$e_q^\wedge = g^\wedge - ZZ'^{-1}(g' + e_q) = (g^\wedge - ZZ'^{-1}g') + ZZ'^{-1}e_q = ZZ'^{-1}e_q \quad (5.5)$$

Bien entendu, l'erreur de quantification restera inchangée aux r points choisis pour le signal g' . Etant donné une certaine erreur de quantification tolérée, il s'agit de trouver parmi toutes les configurations possibles de vecteurs g' une de celles qui permettent de reconstituer g^\wedge avec au plus cette erreur en chaque point. Une fois ce signal g' trouvé, il suffit de coder la position des r parmi N points choisis, et de quantifier les r valeurs de ce signal selon le nombre de bits correspondant à l'erreur de quantification tolérée.

Voyons maintenant comment on peut appliquer cette analyse à notre problème de quantification de coefficients polynomiaux. Vu les caractéristiques basses fréquences des fonctions polynomiales sur des régions de grande taille ainsi que la forme géométrique des surfaces engendrées par ces polynômes, il nous a paru raisonnable d'admettre que l'on pouvait obtenir de bons résultats en choisissant les r points de g' suffisamment éloignés. Ainsi, l'erreur de quantification devrait rester acceptable en choisissant $r-1$ points uniformément répartis sur la frontière de la région considérée. Le dernier point peut être défini lui par le point plus proche du centre de gravité de la région et appartenant à cette région. Le signal luminance étant quantifié à 8 bits, il faudra $8r$ bits pour coder les r coefficients du polynôme d'approximation. Il est probablement possible de trouver une quantification plus efficace en adoptant la démarche proposée précédemment. Il suffirait de chercher s'il existe r points de la région pour lesquels on puisse quantifier les niveaux de gris du signal g' correspondant avec moins de 8 bits, tout en n'ayant pas à transmettre trop d'information pour décrire la position de ces points. Vu la part peu importante du codage de l'information texture, nous nous sommes satisfait du premier résultat.

Analysons maintenant s'il est possible de regrouper les configurations polynomiales de différentes régions, de manière à réduire ultérieurement la redondance. En effet, si l'algorithme de rassemblement a permis d'éliminer bon nombre de régions de l'image divisée en regroupant toutes les régions adjacentes qui pouvaient être décrites par une même fonction d'approximation, il s'agit d'étudier s'il existe des régions séparées par une région correspondant à un autre espace régulier de l'image pouvant être décrites par le même polynôme d'approximation. Une manière d'y parvenir consiste à appliquer le processus de rassemblement à des régions non nécessairement adjacentes. On a préféré

comparer les coefficients polynomiaux. Une façon élégante de le faire consiste à utiliser la quantification vectorielle.

5.2.2 Quantification vectorielle des coefficients polynomiaux:

La quantification vectorielle se différencie de la quantification scalaire par le fait qu'elle est effectuée simultanément pour toutes les composantes du vecteur de coefficients plutôt que pour chaque coefficient pris séparément. Les performances élevées de cette technique ont été montrées dans le contexte du codage de la parole et de la reconnaissance des formes appliquée à la parole [67,68]. Récemment, elle a trouvé des adeptes dans le codage d'images [69,70] et de séquence d'images [71]. Ainsi qu'il a été montré par Makhoul et al. [67], la quantification vectorielle permet de tenir compte de la dépendance linéaire et non linéaire (au sens statistique), de la densité de probabilité et de la dimension du vecteur aléatoire à quantifier. Voyons d'une manière générale la formulation du problème de quantification vectorielle.

Formulation du problème:

Considérons un ensemble U de vecteurs $u = [u_1 \ u_2 \ \dots \ u_r]^T$ de dimension r dont les composantes constituent des variables aléatoires réelles. La quantification vectorielle est l'opération qui consiste à transformer ce vecteur u en un autre vecteur de même dimension noté v . Le vecteur v est appelé le vecteur de reconstruction associé au vecteur u . Il fait partie d'un ensemble fini de L vecteurs $v_i = [v_{i1} \ v_{i2} \ \dots \ v_{ir}]$, noté $V = \{v_i, i=1, \dots, L\}$. De manière à obtenir l'ensemble V , on va décomposer l'espace à r dimensions du vecteur aléatoire u en L régions ou classes $\{C_i, i=1, \dots, L\}$ et associer à chaque classe un vecteur v_i . La figure 5.13 donne un exemple de décomposition d'un espace bidimensionnel en 14 classes.

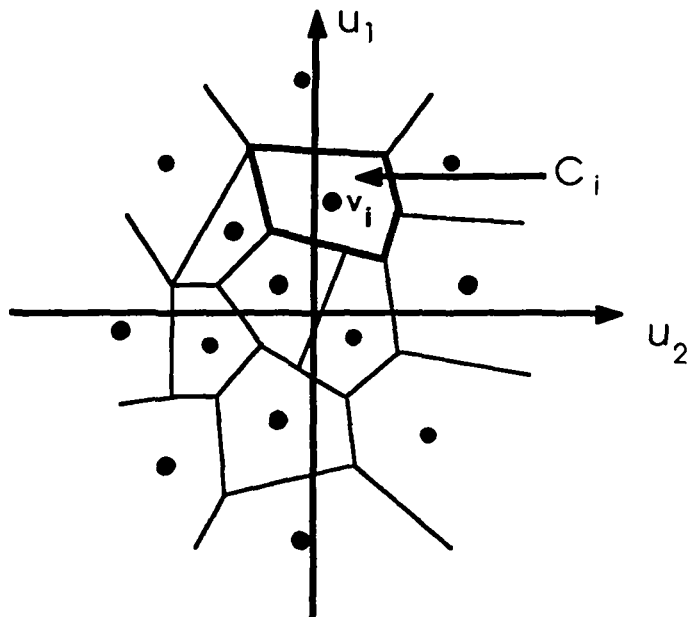


Figure 5.13: Subdivision d'un espace bidimensionnel ($r=2$) en $L=14$ classes. Tous les vecteurs u situés à l'intérieur d'une région définissant une classe C_i seront

quantifiés à la valeur v_i . La forme des différentes régions peut être quelconque. Elle dépend de la mesure de distorsion choisie.

Lorsque le vecteur u est transformé en un vecteur v par quantification, on commet une erreur de quantification et une mesure de distorsion $d(u,v)$ doit être introduite. Nous discuterons quel genre de mesure on peut utiliser dans le contexte des vecteurs de coefficients polynomiaux qui servent à décrire la luminance de chaque région. Analysons d'abord comment obtenir pour une mesure de distorsion choisie l'ensemble V des vecteurs de reconstruction v_i .

Algorithmes de coalescence:

Une méthode permettant d'obtenir l'ensemble V des L vecteurs de quantification est basée sur l'application de l'algorithme du "K-means" [72]. Cet algorithme doit viser à une minimisation des mesures de distorsion sur l'ensemble des vecteurs u à quantifier. Il effectue une coalescence en L classes de l'espace de M vecteurs u servant à l'apprentissage. Il fonctionne itérativement de la manière suivante (m constitue le numéro de l'itération):

1.- mettre m à 0. Choisir un ensemble adéquat de L vecteurs initiaux $v_i(0)$.

2.- Classer l'ensemble des M vecteurs d'apprentissage selon la règle du plus proche voisin:

$$u \in C_j(m) \text{ ssi } d(u, v_j(m)) \leq d(u, v_i(m)) \quad \forall i \neq j \quad (5.6)$$

3.- mise à jour des vecteurs de reconstruction ($m \leftarrow m+1$) en calculant pour chaque classe le vecteur moyenne de tous les vecteurs d'apprentissage u qui ont été associés à cette classe.

4.- si la différence entre une mesure globale de distorsion à l'étape m et celle que l'on avait à l'étape $m-1$ passe au-dessous d'un certain seuil, arrêter le processus sans quoi reprendre l'algorithme au point 2.

On peut prouver que cet algorithme converge vers un minimum local [73]. La solution en général obtenue n'est pas unique. On peut essayer de trouver un optimum global en prenant plusieurs valeurs initiales pour les vecteurs de reconstruction $v_i(0)$ et en conservant la solution qui aboutit au minimum de distorsion. Il n'existe pas de méthode générale pour effectuer le point 1 de l'algorithme surtout si l'on désire subdiviser l'espace du vecteur aléatoire u en un nombre variable de classes.

Aussi, nous avons choisi un autre algorithme de coalescence inspiré de la technique adaptative de rassemblement, correspondant à une technique de classification ascendante hiérarchique. La première étape consiste à construire un graphe dont les noeuds représentent toutes configurations possibles de vecteurs de coefficients. Ainsi, à partir d'une image segmentée en 99 régions on construit un graphe formé de 99 noeuds. Les branches du graphe servent à lier chaque noeud 2 à 2 de telle sorte que l'on puisse associer les vecteurs de coefficients de deux régions quelconques de la segmentation. De ce fait, un graphe constitué de M noeuds possède $M(M-1)/2$ branches. Pour l'exemple précédent, le graphe est

formé de 4851 branches. Une fois ce graphe réalisé, on associe à chacune de ces branches la mesure de distorsion qui résulterait du regroupement des noeuds qu'elles joignent. On recherche alors la branche présentant le minimum de distorsion et on rassemble les deux noeuds qu'elle relie. On calcule alors les nouvelles distorsions qui existent entre le noeud ainsi créé et le reste des $N-2$ noeuds du graphe. Le processus de rassemblement est alors itéré jusqu'à atteindre L noeuds correspondant aux L vecteurs de reconstruction désiré.

Cet algorithme possède deux avantages par rapport au K-means:

- 1.- il permet d'atteindre l'optimum global pour la mesure de distorsion choisie.
- 2.- il donne directement la forme des classes C_i et des vecteurs de reconstruction v_i pour toutes sortes de valeurs L .

Néanmoins, pour un large nombre M de vecteurs d'apprentissage, il peut nécessiter des temps de traitement relativement élevés, sans parler de problèmes de place mémoire nécessaire. En effet, pour construire le graphe initial, il faut calculer $M(M-1)/2$ mesures de distorsion. La première étape de rassemblement nécessite de recalculer $M-2$ mesures de distorsion, la deuxième $M-3$ et ainsi de suite jusqu'au $M-L-1$ rassemblement, si bien qu'il aura fallu calculer au total N mesures de distorsion données par :

$$\begin{aligned}
 N &= M(M-1)/2 + \sum_{m=M-((M-L-1)+1)}^{M-2} m \\
 &= M(M-1)/2 + (M-1)(M-2)/2 - L(L-1)/2 \\
 &= (M-1)^2 - L(L-1)/2
 \end{aligned} \tag{5.7}$$

Pour effectuer les comparaisons des différentes mesures de distorsion, on peut utiliser la structure d'arbre binaire balancé proposée au chapitre 4. La génération de l'arbre requiert au départ $O(M^4 \log^2 M)$ opérations. Cependant, dans notre application le nombre de configurations de vecteurs de coefficients polynomiaux reste relativement réduit puisque l'on considère des segmentations ne dépassant pas les 200 régions, ce qui justifie l'intérêt de cet algorithme de coalescence. Il s'agit maintenant de définir les mesures de distorsion permettant d'associer différentes configurations de coefficients.

Mesures de distorsions:

Pour être utile, une mesure de distorsion doit pouvoir être analysée et calculée. Elle doit rendre compte d'un point de vue subjectif des distorsions causées au signal étudié, en l'occurrence une image. A partir de deux vecteurs de coefficient u_1 et u_2 correspondant respectivement à deux régions quelconques de la segmentation, on peut ainsi calculer l'approximation polynomiale optimale au sens des moindres carrés sur l'union des deux régions définie par:

$$g_q = Zv \tag{5.8}$$

où v est simplement donné par la relation (4.7) où on a remplacé u par v . g_q est à comparer aux approximations avant regroupement (mesure d'erreur courante)

ou au signal original à l'intérieur de l'union des deux régions. Une mesure de distorsion qui tient parfaitement compte du signal à approximer peut être donc basée sur l'un des critères EQ, EQC, EQCP ou EQP proposés au chapitre 4. (le critère EQM est à écarter pour les raisons évoquées dans ce chapitre). Il nous a cependant semblé que l'on ne fait que généraliser ainsi la procédure adaptative de rassemblement à des régions non forcément contiguës. En outre, le calcul de ces mesures de distorsions est relativement coûteux (voir section 4.4).

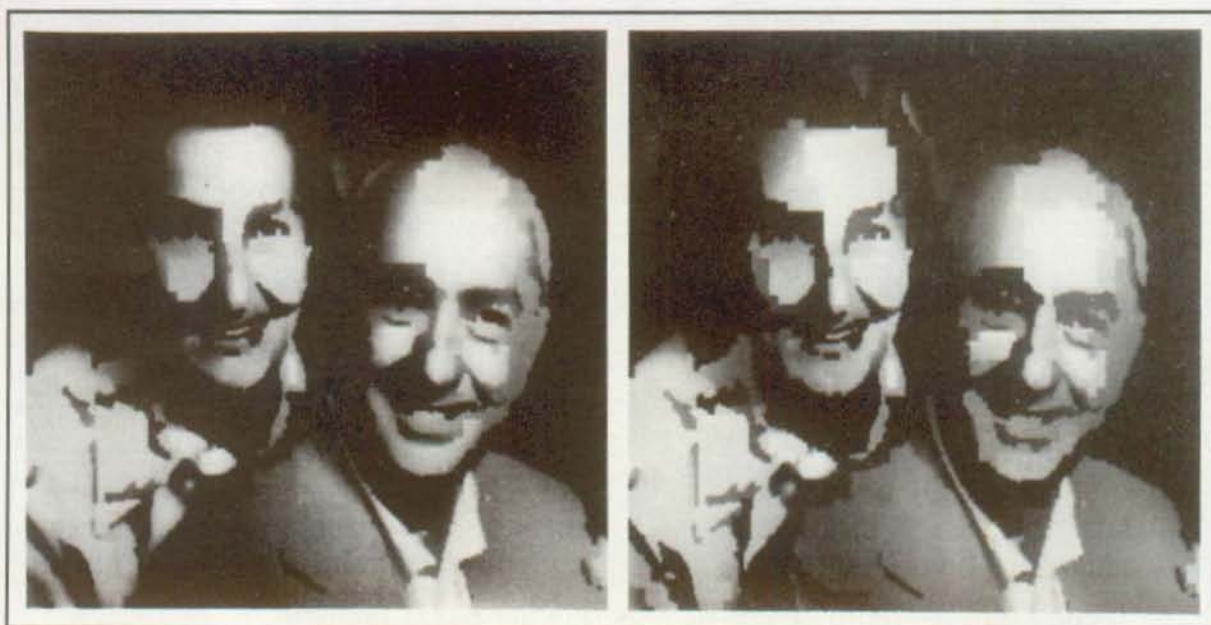
On a ainsi préféré baser la comparaison uniquement sur la valeur des vecteurs de coefficients. En effet, ils se rapportent à des régions de taille relativement importante si bien qu'ils constituent un bon modèle du signal luminance de ces régions si la qualité de la segmentation peut être jugée satisfaisante. Les coefficients d'approximation permettent de caractériser chaque région quant à sa luminance indépendamment du nombre de points de cette région.

A première vue, on peut utiliser une mesure de distorsion de type erreur quadratique entre vecteurs de coefficients donnée par la relation:

$$d(u,v) = (u-v)^T(u-v) \quad (5.9)$$

Comme cette mesure sert à estimer la distorsion qui résulterait du regroupement de deux régions quelconques de la segmentation (voir algorithme de coalescence), on a décidé de comparer les coefficients u_1 et u_2 respectifs par rapport à leur centre de gravité. Si l'on décide d'associer deux vecteurs u_1 et u_2 , le vecteur résultant u_{12} est exprimé par la moyenne de ces vecteurs pour poursuivre l'algorithme de coalescence. Si un nouveau vecteur u_3 est associé à u_{12} , le vecteur résultant s'obtient par moyennage des 3 vecteurs u_1 , u_2 et u_3 , de manière à tenir compte du nombre de régions qui forment la classe correspondante. Une fois l'algorithme de coalescence terminé, toutes les régions dont les vecteurs de coefficients font partie d'une certaine classe C_i sont décrites par un vecteur de reconstruction v qui correspond à la solution optimale au sens des moindres carrés du polynôme sur l'ensemble de ces régions. On garantit ainsi la correspondance optimale entre le signal luminance de ces régions et les coefficients du polynôme servant à les décrire.

La figure 5.14a illustre l'image du couple reconstituée pour la segmentation formée de 99 régions de la figure 5.6 dont on a réduit le nombre de vecteurs de coefficients à 80 selon la procédure que l'on vient de décrire. Il n'apparaît pas de distorsions importantes. Il ne faut cependant pas poursuivre le processus de coalescence outre mesure car la quantification vectorielle ne fonctionne correctement que pour des régions à variation très lente. L'association de régions éloignées peut détruire la variation de luminance apparaissant dans ces régions. Les distorsions qui apparaissent sont alors considérables ainsi qu'on peut le voir à la figure 5.14b où l'on n'a plus que 50 vecteurs de reconstruction. Les résultats sont nettement inférieurs à la segmentation formée de 49 régions que l'on obtient à la fin de l'algorithme adaptatif de division et rassemblement (voir figure 4.17a). En effet, lors du processus de quantification vectorielle, on ne tient absolument pas compte du voisinage des régions.



a) 80 vecteurs de reconstruction

b) 50 vecteurs de reconstruction

Figure 5.14: Image du couple à 99 régions avec vecteurs de coefficients quantifiés par erreur quadratique en 80 et 50 configurations différentes.

Les défauts qui résultent d'une mesure d'erreur quadratique proviennent essentiellement du fait que les coefficients correspondant à des monômes de degré élevé ont des valeurs beaucoup plus faibles que les autres coefficients. Ainsi, pour un polynôme de degré 3, le terme x^3 a une dynamique très réduite par rapport au terme constant même si l'on exprime le polynôme par rapport au centre de gravité de la région. Les associations qui en résultent au moment de l'application de l'algorithme de coalescence peuvent ainsi être faussées. On peut normaliser chaque vecteur de coefficients en tenant compte de sa valeur moyenne et de sa variance. Cette normalisation peut être réalisée par l'utilisation de la distance de Mahalanobis définie par:

$$d_m(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T \Gamma^{-1} (\mathbf{u} - \mathbf{v}) \quad (5.10)$$

où la matrice Γ correspond à la matrice de covariance et \mathbf{u}_m est le vecteur moyenne:

$$\Gamma = E((\mathbf{u} - \mathbf{u}_m)(\mathbf{u} - \mathbf{u}_m)^T) \quad \text{et} \quad \mathbf{u}_m = E(\mathbf{u}) \quad (5.11)$$

L'application des formules (5.10) et (5.11) peut se faire indépendamment du choix d'une origine globale pour le calcul des vecteurs de coefficient \mathbf{u} (ceci est démontré dans l'annexe H). Les vecteurs de coefficients ont été cependant définis de façon locale, relativement au centre de gravité des régions auxquelles ils se rapportent. Après chaque association, le nouveau vecteur est obtenu par moyennage comme dans le cas de la mesure d'erreur quadratique. A la fin de l'algorithme de coalescence, les vecteurs de reconstruction sont calculés par minimisation de l'erreur quadratique.

La figure 5.15a illustre l'image du couple reconstituée pour la segmentation formée de 99 régions de la figure 5.6 dont on a réduit le nombre de vecteurs de coefficients à 80 en utilisant la distance de Mahalanobis comme mesure de distorsion. La figure 5.15b montre la même image reconstituée si l'on n'a plus que 50 vecteurs de reconstruction. Les qualités des figures 5.15 et 5.14 sont comparables pour un même nombre de vecteurs de reconstruction. Ceci confirme l'hypothèse qu'on ne peut réduire outre mesure le nombre de configurations polynomiales par quantification vectorielle. Les résultats obtenus en utilisant la distance de Mahalanobis ne sont pas nécessairement meilleurs sur le plan visuel que ceux de la figure 5.14 (erreur quadratique).



a) 80 vecteurs de reconstruction

b) 50 vecteurs de reconstruction

Figure 5.15: Image du couple à 99 régions avec vecteurs de coefficients quantifiés par distance de Mahalanobis en 80 et 50 configurations différentes.

Les configurations de vecteurs de coefficients seront cependant calculées sur la base de la mesure de distorsion (5.10) en raison de son effet de normalisation sur les différents coefficients. Une fois les classes ainsi définies, les vecteurs de reconstruction \mathbf{v}_i sont estimés pour chaque classe en calculant l'approximation polynomiale optimale au sens des moindres carrés sur l'ensemble des régions qui se rapportent à cette classe. L'approximation obtenue définit ce que l'on appelle la surface de reconstruction de la classe C_i .

5.2.3 Variation du nombre de coefficients par région:

Une fois les surfaces de reconstruction connues, il s'agit d'analyser si on peut les classer par degrés de polynôme. Une image segmentée au moyen de polynômes de degré 3 peut comprendre un certain nombre de surfaces de reconstruction sans variation (degré 0); il peut y avoir d'autres surfaces de reconstruction à variation linéaire (degré 1), quadratique (degré 2) ou cubique (degré 3). Chaque type de polynôme peut être codé moyennant 8 bits par

coefficient (voir paragraphe 5.2.1) ce qui fait varier le coût d'une région quant à son information texture de 8 à 80 bits selon le genre de variation qui la caractérise.

Supposons d'avoir effectué la segmentation avec un polynôme de degré α . Pour estimer le degré de la surface de reconstruction d'une classe C_i , on calcule toutes les approximations polynomiales de degré 0 à $\alpha-1$ optimales au sens des moindres carrés sur le domaine de l'image se rapportant à C_i . On choisit ensuite l'approximation qui est la moins coûteuse en nombre de bits mais dont l'erreur quadratique moyenne par rapport à l'approximation de degré α reste inférieure à un certain seuil T .

Une fois la classification des surfaces de reconstruction réalisée, on estime la statistique de chaque degré de surface. On peut alors coder de façon optimale par Huffman le genre de variation (nulle, linéaire, quadratique ou cubique) associée à chaque surface de reconstruction. En relation avec le genre de variation, on transmet les 1, 3, 6 ou 10 coefficients nécessaires à la reconstruction. La figure 5.16 illustre l'image du couple de la figure 5.15a (80 surfaces de reconstruction) pour deux valeurs différentes du seuil T permettant de déterminer l'importance des termes de chaque degré. N_0 à N_3 représentant le nombre de surfaces de chaque degré.



a) $T=5^2$, $N_0=1$, $N_1=0$, $N_2=79$, $N_3=0$

b) $T=15^2$, $N_0=12$, $N_1=17$, $N_2=51$, $N_3=0$

Figure 5.16: Image du couple à 99 régions avec 80 surfaces de reconstruction classés par type de variation.

5.2.4 Stratégie globale:

Pour coder aussi efficacement que possible l'information texture, il faut:

1.- effectuer une quantification vectorielle des vecteurs de coefficients (paragraphe 5.2.2) de manière à pouvoir associer les parties disjointes de l'image

qui peuvent être caractérisées par un même polynôme d'approximation, c'est-à-dire une même surface de reconstruction. Vu le faible nombre de régions après application de l'algorithme de segmentation adaptative, il faut prendre soin de n'associer qu'un faible nombre de vecteurs de coefficients.

2.- classer les différentes surfaces de reconstruction selon le genre de variation qui les caractérisent (paragraphe 5.2.3).

3.- transmettre le degré du polynôme caractérisant chacune des L surfaces de reconstruction. Les quatre symboles caractérisant les quatre degrés de polynômes sont codés selon Huffman.

4.- transmettre autant de pixels par surface de reconstruction que de nombre de coefficients polynomiaux qui la caractérise. Ces pixels sont quantifiés à 8 bits. Ils sont choisis de façon systématique une fois le domaine associé à la surface de reconstruction connu (voir paragraphe 5.2.1). Les coefficients du polynôme sont obtenus par interpolation de ces pixels.

5.- Définir les domaines associés à chaque surface de reconstruction. Le codage des contours permet de définir la forme des différentes régions. Il faut encore adresser toutes les régions caractérisées par une même surface de reconstruction à l'un des L lots possibles de coefficients. En effet, à cause de l'élimination des liens de régions et du regroupement des vecteurs de coefficients, une surface de reconstruction peut être associée à un ensemble de régions disjointes. Vu les valeurs en général comparable du nombre de parties disjointes et du nombre de surfaces de reconstruction (rapport de 1 à 2), il est inutile de trouver le code de Huffman à partir de la statistique du nombre de régions associées à une certaine surface de reconstruction. Supposons de devoir représenter M régions disjointes et que a représente la plus petite puissance de 2 supérieure à L . On peut lier chaque région à un vecteur de coefficient par $\log_2 a$ bits. On peut également lier chaque surface de reconstruction à une région de l'image avec un mot de $\log_2 b$ bits si b représente la plus petite puissance de 2 supérieure à M . Comme il faut créer M liens avec $M \geq L$, la liaison entre la partition de l'image et les surfaces de reconstruction est donc réalisée avec au plus $Mx\log_2 a$ bits.

Dans l'exemple de la figure 5.16a, il faut 3880 bits pour coder les 80 surfaces de reconstruction. Il en faut 3061 pour les 80 surfaces de reconstruction de la figure 5.16b. Pour adresser les 138 parties disjointes de la partition avec l'une de ces 80 surfaces, on aura besoin de $138 \times 7 = 966$ bits supplémentaires.

Après avoir défini la stratégie de codage de l'information segmentée, nous allons proposer des méthodes permettant d'éliminer des artéfacts de l'image reconstruite.

5.3 Elimination des faux contours :

Les faux contours de l'image reconstruite peuvent apparaître aux frontières des régions qui n'épousent pas des contours réels de l'image originale. En effet, si le signal présente un gradient important perpendiculairement à la direction de cette frontière, on a créé un faux contour qui n'existait pas dans l'image de départ. Ce phénomène est relativement fréquent à cause de la propriété qu'ont les approximations polynomiales de s'éloigner du signal original sur le bord du domaine où elles sont estimées. Si ce phénomène ne présente aucun désavantage pour les frontières des régions correspondant à des contours réels de l'image, il en est tout à fait autrement pour les autres frontières.

Pour s'en rendre compte, il suffit d'observer les partitions obtenues par l'algorithme de division sans utilisation d'une image de contrôle des contours (voir figures 3.14 et 3.16). Ce même type d'artéfact est bien visible à la fin de l'algorithme de division et rassemblement. Il suffit d'observer le front de la femme dans l'image du couple de la figure 5.16.

Pour faire disparaître les faux contours, il faut appliquer localement aux frontières définissant ces faux contours un filtre ayant une caractéristique de type passe-tout dans la direction parallèle à la frontière et passe-bas dans la direction orthogonale de manière à couper la transition artificielle de luminance.

Une autre possibilité qui a été retenue ici consiste à appliquer un algorithme de lissage localement au faux contour. Les caractéristiques de lissage peuvent être formulées d'une façon analogue au problème d'optimisation sous-contrainte proposé par Carlsson [74]. A partir d'une estimation locale $F(g_x, g_y, g_{xy}, \dots)$ de la luminance du signal en fonction des coordonnées de l'image g dont la dérivée selon x est notée g_x , celle selon y g_y , la dérivée seconde selon x puis y $g_{xy} \dots$, l'image rehaussée $g_r(x, y)$ est obtenue en minimisant la variation globale:

$$\min_g \iint F(g_x, g_y, g_{xy}, \dots) dx dy \quad (5.12)$$

en satisfaisant les conditions aux limites imposées aux bords de la région à lisser. Si l'on utilise comme mesure de variation, le carré du module du gradient, ce problème d'optimisation est équivalent à résoudre l'équation de Laplace:

$$\nabla^2 g_r = 0 \quad (5.13)$$

Cette équation peut être résolue itérativement par relaxation [75] :

$$g_r^{(i+1)}(k, l) = g_r^{(i)}(k, l) + (w/4) \nabla^2 g_r^{(i)}(k, l) \quad (5.14)$$

où l'estimation du Laplacien de g_r à la position (k, l) est donnée par:

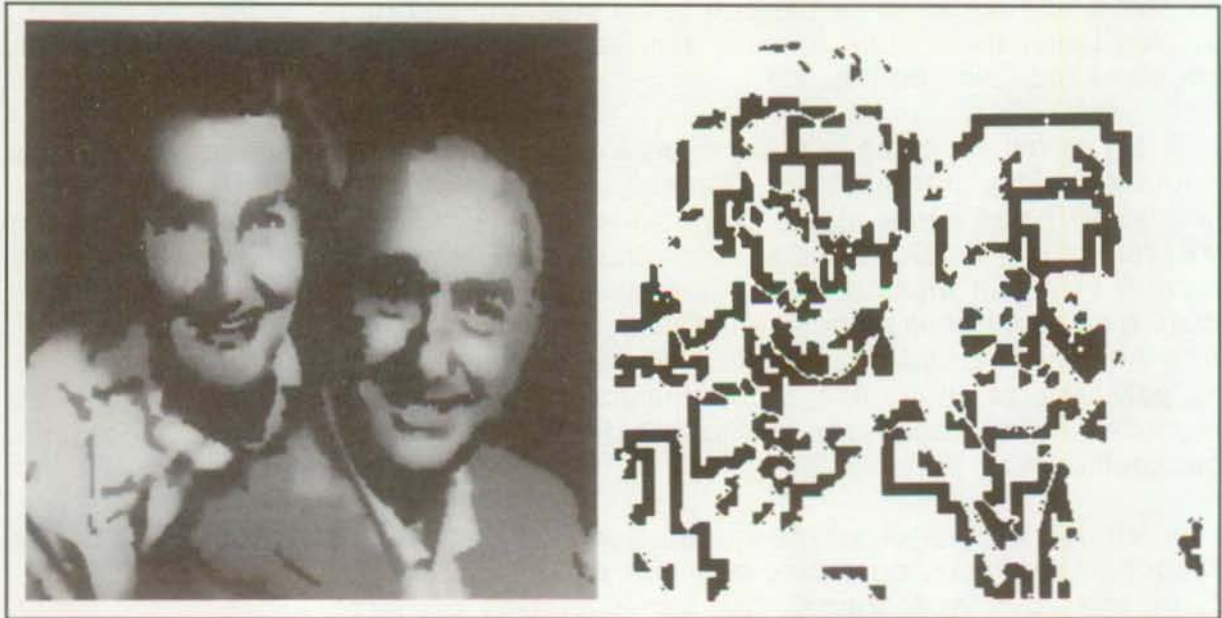
$$\begin{aligned} \nabla^2 g_r^{(i)}(k, l) &= g^{(i+1)}(k-1, l) + g^{(i+1)}(k, l-1) + g^{(i)}(k+1, l) + g^{(i)}(k, l+1) \\ &\quad - 4g^{(i)}(k, l) \end{aligned} \quad (5.15)$$

w définit le paramètre de relaxation. Il sert à déterminer la stabilité et la vitesse de convergence de l'algorithme. Les conditions aux limites imposent $g_r(k,l)=g(k,l)$ en ces points limites. Il reste à déterminer comment définir les limites des régions à lisser.

La première étape consiste à trouver les faux contours de l'image reconstruite après segmentation. Ceci peut être réalisé avec l'aide de l'image de contrôle des contours. On compare celle-ci au graphe de contours de la section 5.1. Pour chaque point de contour de ce graphe, on analyse s'il existe dans son voisinage un point contour de l'image de contrôle correspondante. Si tel est le cas, le point en question correspond à un contour réel. La fenêtre de voisinage tolère les distorsions éventuelles qui pourraient exister entre contours réels et frontières de régions. Vu les exigences imposées à la segmentation, on a limité la taille de cette fenêtre à 3×3 .

La deuxième étape consiste à définir la bande qui doit être lissée autour des faux contours que l'on vient de détecter. Cette bande est obtenue en plaçant en chaque point constituant un faux contour une fenêtre de taille supérieure à la précédente mais non trop large pour éviter d'effectuer le lissage sur une portion trop importante de l'image reconstruite et donc de dégrader l'approximation polynomiale. Aussi, avons-nous choisi de fixer cette taille à 5×5 . Si cette fenêtre ne contient aucun contour, la région qu'elle occupe sera lissée. En revanche, la présence d'un contour justifie de séparer les portions correspondantes de la fenêtre 5×5 situées d'un même côté du contour.

Le résultat de l'algorithme de lissage ainsi décrit est montré à la figure 5.17a pour l'image du couple de la figure 5.16a. La figure 5.17b indique les zones de l'image de la figure 5.16a qui ont été lissées.



a) Image de la figure 5.16a rehaussée

b) Portions lissées (en noir)

Figure 5.17: Image rehaussée du couple à partir d'une segmentation formée de 99 régions (80 surfaces de reconstruction).

Il est important de remarquer que l'application de l'algorithme de lissage au niveau de l'image reçue nécessite en général la transmission des portions de frontières correspondant à de faux contours. Pour cela, il suffit d'indiquer lors de la transmission de la chaîne des points constituant le graphe de contours, ceux qui correspondent à des débuts et à des fins de faux contours dépassant une certaine longueur (il est inutile de coder des portions trop courtes). Il est en général moins coûteux de transmettre les portions de frontières correspondant à de faux contours que celles correspondant à des contours réels, car la plupart des frontières de régions constituent des contours réels. Ainsi, en admettant que l'on transmette au plus 16000 points de contour pour coder le graphe de contours, il faut au plus 14 bits par point constituant une extrémités de contour. En réalité, en transmettant la différence d'adresse entre ces points, on peut ramener ce coût à 12 bits par point. Une autre possibilité eut été de définir un nouveau symbole au niveau du codeur de graphe de contour pour indiquer ces points d'extrémités de faux contours. Il existe 79 points de ce genre reliant des faux contours d'une longueur supérieure à 20 points dans l'image de la figure 5.17. Aussi, faut-il environ 948 bits pour transmettre au récepteur les faux contours à éliminer par lissage.

5.4 Conclusion :

Dans ce chapitre, nous avons abordé le problème de la représentation de l'information contour et de l'information texture découlant de la segmentation adaptative par division et rassemblement. Deux méthodes séparées ont été développées pour coder chaque type d'information. On a étudié ensuite comment il était possible de rehausser la qualité des images reconstruites après codage par l'application d'un algorithme de lissage local. Cet algorithme sert à éliminer les faux contours pouvant apparaître à la frontière de certaines régions. A partir d'une zone à lisser, on cherche à minimiser les variations du gradient de l'image rehaussée en résolvant une équation de Laplace selon une méthode de relaxation. Les valeurs sur les bords de la zone à lisser définissent les conditions aux limites que doit satisfaire l'équation de Laplace.

En ce qui concerne le codage, on a décrit comment les frontières de régions pouvaient être parfaitement reconstruites à partir du codage de l'image interfrontière liée à une segmentation sans liens de régions. Le code utilisé permet d'éviter toute distorsion dans la reconstruction de cette image interfrontière. On a montré qu'il faut environ 1.5 bit par point frontière. Le codage de l'information texture se fait en trois étapes: quantification vectorielle des vecteurs de coefficients polynomiaux en L classes; classification des L vecteurs de reconstruction par degré du polynôme selon la variation du domaine de l'image auquel chaque vecteur se rapporte; enfin, codage des coefficients des polynômes en utilisant au plus 8 bits par coefficient.

Vu le nombre élevé de points contours, on a vu sur un exemple que l'information contour représente au moins deux tiers de l'information totale à coder. Cette remarque va être confirmée par le chapitre qui suit où sont présentés les résultats de la méthode.

6. Résultats en codage d'images

Ce chapitre présente les résultats de la méthode de segmentation adaptative pour le codage d'images. La méthode a été appliquée aux trois images originales présentées au chapitre 2: image du bâtiment (figure 2.2), image du caméraman (figure 2.3) et image du couple (figure 2.4). On discute l'effet des différents paramètres importants permettant de modifier la forme de la segmentation finale et de l'image reconstruite correspondante, notamment le degré du polynôme d'approximation et le nombre de régions formant la segmentation. Les autres paramètres prennent sauf indication contraire les valeurs par défaut indiquées aux chapitres 3 à 5. Si cela n'a été fait, leurs valeurs seront précisées au cours de la présentation.

Ce chapitre comprend quatre sections. Dans la première, on présente tous les paramètres intervenant dans l'image reconstruite. Le calcul du facteur de compression γ est également expliqué. Dans la seconde, les résultats de codage des trois images sont présentés et discutés pour plusieurs degrés du polynôme d'approximation et du nombre de régions de la segmentation. La troisième section discute de l'effet du prétraitement de l'image originale. Enfin, la dernière section propose des améliorations de la méthode en jouant au niveau de la méthode de segmentation et de la composante aléatoire du signal reconstruit. Dans le premier cas, on propose de commencer la segmentation par des polynômes de degré 0 puis de passer à des polynômes de degré plus élevé pour arriver à la segmentation finale. Dans le second, on explique comment estimer une composante aléatoire relativement à chaque région.

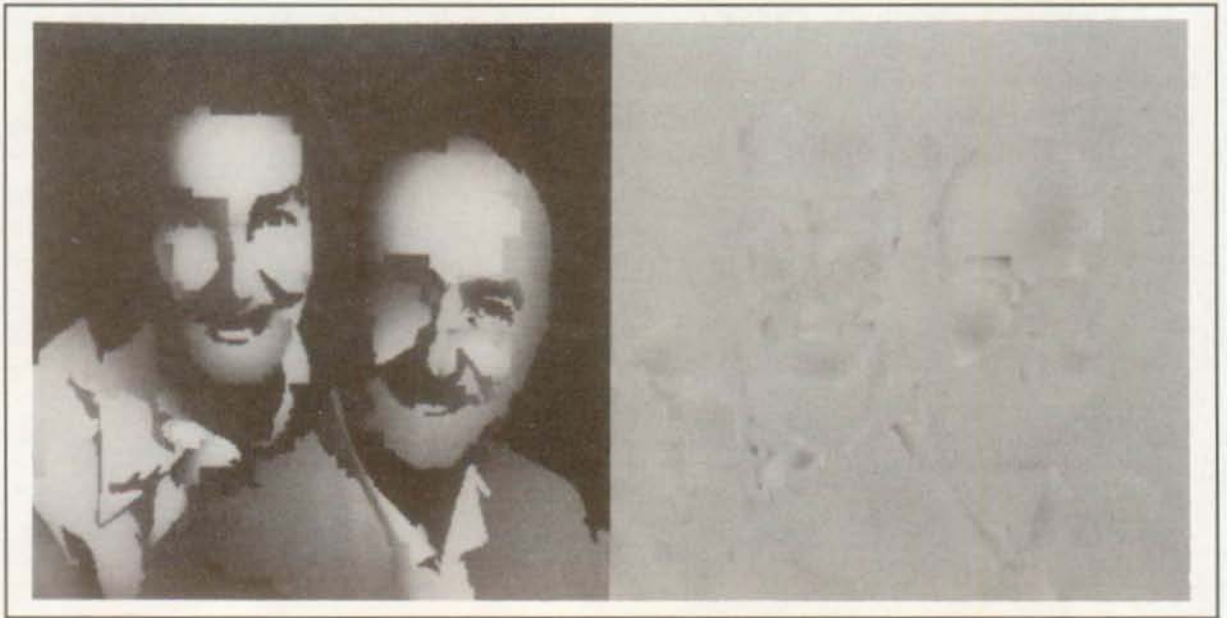
6.1 Paramètres d'expérimentation et calcul de compression :

Définir une procédure générale d'expérimentation pour toutes les valeurs possibles des paramètres de l'algorithme de codage est pratiquement impossible à cause de son explosion combinatoire. Ceci est d'autant plus vrai que l'effet de certains paramètres peut être différent d'une image à l'autre. Certains de ces paramètres sont heureusement assez indépendants les uns des autres pour que l'on n'ait pas besoin de juger de leur effet conjugué pour fixer leur valeur. Ainsi, la taille d'un faux contour à éliminer ne dépend pas vraiment de l'ordre du polynôme d'approximation à partir du moment où la frontière correspondante a été admise comme définissant un faux contour. La présence d'un faux contour est par contre indirectement dépendante du degré du polynôme car celui-ci aura une influence sur la forme de la segmentation et donc l'apparition du faux contour en question. Les paragraphes qui suivent illustrent comment les valeurs des différents paramètres d'expérimentation ont été choisies et sur quelle base le facteur de compression est estimé.

6.1.1 Degré du polynôme:

Les trois images traitées seront présentées pour les 4 valeurs des degrés des polynômes d'approximation (0, 1, 2 et 3). On verra que ce paramètre peut avoir des effets différents d'une image à l'autre. Le degré du polynôme sera noté α . A première vue, il semblerait qu'on puisse obtenir des résultats de qualité comparable pour des polynômes de degré 2 et 3 pour un même nombre de

régions, vu la correspondance qui existe déjà entre des polynômes de degré 2 et des parties d'objets de scènes naturelles. En réalité, une augmentation du degré du polynôme permet de suivre des surfaces à variation très complexe. Il suffit pour cela d'observer la figure 4.16. La région correspondant à la joue droite de l'homme présente de multiples variations. Une telle précision n'aurait pu être obtenue pour des polynômes de degré 2 qu'en séparant les régions correspondantes au niveau de la segmentation. Pour le prouver, on montre à la figure 6.1 le résultat de la figure 4.16 où toutes les régions ont été approximées par des polynômes de degré 2. La figure 6.1a illustre l'image reconstruite correspondante. La figure 6.1b montre la différence entre les deux images. On remarque l'erreur apparaissant notamment dans le coin inférieur droit de la région précédemment décrite. Les résultats présentés à la section 6.2 montreront d'autres effets du degré du polynôme d'approximation.



a) Reconstruction avec $\alpha=2$.

b) Différence avec l'image de la fig. 4.16.

Figure 6.1: Segmentation de l'image du couple en 99 régions en utilisant des polynômes de degré 3 (voir fig. 4.16) reconstruite avec des polynômes de degré 2.

6.1.2 Nombre de régions:

Ce paramètre est crucial car il contrôle directement la qualité et le facteur de compression associés à une image segmentée. Lors de la présentation des résultats, il sera noté N_r . Les autres paramètres étant fixés, la diminution du nombre de régions entraîne systématiquement une baisse de la qualité (au moins, une perte de détail) et une augmentation du facteur de compression. En effet une diminution du nombre de régions d'une unité entraîne la disparition de la portion de frontière commune aux deux régions que l'on a rassemblées; de plus, il existe une région de moins à laquelle on doit associer une surface de reconstruction. Les images seront présentées pour trois à quatre valeurs différentes du nombre de régions. Après modification de la segmentation pour éliminer les liens de région facilitant le codage des frontières de régions, une segmentation à N_r régions est formée de M parties disjointes (avec $M \geq N_r$). L'image interfrontière 4-connexe à M

parties disjointes associée à une certaine segmentation sera systématiquement indiquée avec l'indice b sur chaque figure. Le nombre de points frontières correspondant est indiqué par n_c .

6.1.3 Autres paramètres intervenant dans la qualité des résultats:

Lors de la première phase de la segmentation, c'est-à-dire la subdivision de l'image en des carrés de différentes tailles, les paramètres impliqués en dehors du degré du polynôme d'approximation sont le seuil T_1 sur l'erreur quadratique moyenne (EQM) et l'image de contrôle des contours. Vu les remarques faites au chapitre 3 montrant la bonne correspondance entre l'image reconstruite après division et l'image originale à partir d'une bonne image de contrôle des contours indépendamment de la valeur de T_1 , celle-ci sera maintenue égale à 15^2 .

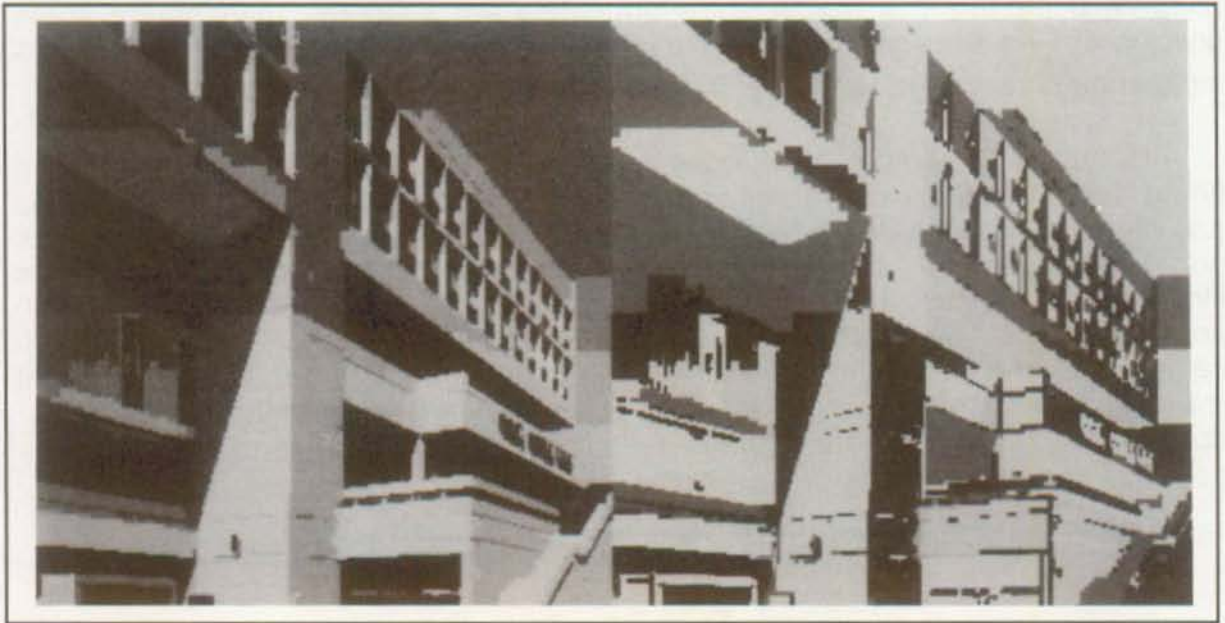
L'effet de l'image de contrôle des contours sur les résultats de la segmentation finale doit être étudié conjointement avec l'image correspondante des structures positives et négatives. Mis à part les résultats discutés à la section 6.3 (les images originales ayant subi un prétraitement), les images de contrôle seront celles indiquées à la section 2.2. En l'occurrence, il s'agit des figures 2.14d, 2.19 et 2.20 pour les images de contrôle des contours associées aux images du couple du caméraman et du bâtiment, respectivement. Les figures 2.23a, 2.24a représentent les images des structures positives et négatives utilisées pour l'image du couple et du bâtiment, respectivement. L'image de la figure 6.2 montre l'image des structures positives et négatives associée à l'image du caméraman.



Figure 6.2: Image des structures positives et négatives du caméraman

Au niveau du rassemblement, on utilise successivement les critères de dissimilarité EQCP (erreur quadratique courante pondérée) et EQC (erreur quadratique courante). Le seuil A sur le nombre de régions restant à rassembler en faisant usage de EQC plutôt que de EQCP est fixé à 499 régions excepté dans le cas de l'image du bâtiment approximée par des constantes. Pour ce degré d'approximation, on n'utilise pas d'image de contrôle des structures positives et

négligables car les carrés contenant des contours correspondent à des pixels. La figure 6.3 montre le résultat d'une segmentation de l'image du bâtiment formée de 499 régions avec l'utilisation du critère EQCP. La figure 6.3a montre l'image reconstruite alors que la figure 6.3b montre la forme de la segmentation avec 5 niveaux de gris. Les multiples détails de l'image du bâtiment ont causé le regroupement de deux régions situées de part et d'autre d'une partie de l'arête supérieure du bâtiment, plutôt que d'associer des régions différentes de faibles tailles à cause de la pondération par le nombre de points de chaque région dans le calcul de EQCP. Pour éviter ce genre de distorsion, il faut fixer le seuil A à une valeur plus élevée. Elle sera fixée pour les expériences de la section 6.2 à 999 régions.



a) Image reconstruite par des constantes b) Forme de la segmentation ($N_r=499$).

Figure 6.3: Segmentation en 499 régions de l'image du bâtiment par des constantes (utilisation de EQCP seul).

Quelle que soit l'image traitée pour des polynômes de degré supérieur à 0, l'insertion au GCR (graphe de contiguïté de régions) des régions contenant des contours s'effectue lorsqu'il ne reste plus que 299 régions à rassembler.

Une fois la segmentation terminée, la qualité des résultats va dépendre encore des paramètres suivants:

- le nombre $L \leq N_r$ de surfaces de reconstruction.
- le seuil T sur l'erreur quadratique moyenne sur chaque région déterminant le degré du polynôme avec lequel on peut effectuer l'approximation d'une certaine région.
- la longueur minimale d en nombre de pixels des frontières correspondant à des faux contours à transmettre.

Le nombre de surfaces de reconstruction obtenues par quantification vectorielle doit se rapprocher du nombre N_r de régions dont est formée la segmentation lorsque ce nombre décroît. En effet, moins il y a de régions dans une image, moins il est probable d'avoir deux régions disjointes pouvant être approximées par un même polynôme.

La forme de la segmentation est obtenue de manière optimale en choisissant à chaque étape de rassemblement les deux régions pour lesquelles l'erreur quadratique allait augmenter le moins. Pour des images segmentées avec des polynômes de degré α , on peut essayer d'utiliser des polynômes de degré supérieur pour représenter chaque région une fois le processus de segmentation terminé. Ceci est confirmé dans l'exemple de la figure 6.4. Celle-ci présente l'image du couple segmentée en 99 régions avec des polynômes de degré 0. La figure 6.4a montre l'image reconstruite par des constantes. La figure 6.4b indique la forme de la segmentation. La figure 6.4c montre l'image reconstruite avec des polynômes de degré 2. La figure 6.4d montre la différence entre les images des figures 6.4a et 6.4c. L'erreur quadratique moyenne entre ces deux images est de 8.2^2 . Entre les deux figures, l'impression visuelle est nettement améliorée.



a) Image reconstruite avec des constantes b) Forme de la segmentation ($N_r=99$).



c) Image reconstruite avec des cubiques ($\alpha=2$) d) Différence entre 6.4a et 6.4b.

Figure 6.4: Image du couple segmentée en 99 régions avec des constantes, successivement approximée par des polynômes de degré 2.

En ce qui concerne le seuil T , ce dernier est fixé relativement bas (de l'ordre de 5^2) de manière à éviter toute distorsion sensible des surfaces de reconstruction pouvant créer l'apparition de faux contours.

La distance minimale d des faux contours à éliminer est fixée à 20 pixels. Ceci évite de coder de l'information supplémentaire lorsque la segmentation est formée de beaucoup de régions, car il est raisonnable de supposer qu'il n'existe pas de faux contours visibles dans ce cas, à cause de la bonne correspondance des fonctions d'approximation à la frontière de chaque région (notamment pour des polynômes de degré élevé).

Maintenant que tous les paramètres de la méthode ont été énoncés, on décrit comment le facteur de compression peut être calculé.

6.1.4 Facteur de compression:

Dans tous les calculs, on négligera le coût associé au codage de la table de Huffman à cause du nombre réduit de symboles à coder (le coût correspondant est négligeable par rapport à l'information totale à coder).

Pour chaque image codée, il faut dissocier le coût associé à l'information contour noté C_c de celui associé à l'information de texture noté C_t . On appellera C_r le coût associé à la transmission des faux contours nécessaires au rehaussement. Le coût global associé à une certaine image est donné par:

$$C = C_c + C_t + C_r \quad (6.1)$$

Le coût associé à l'information contour C_c dépend du nombre de points du graphe de contours 8-connexe et de la statistique des symboles 1 à 13 de la figure 5.11 associée à ce graphe. Il faudra tenir compte du nombre de parties connexes du graphe noté p . Le coût C_c est obtenu à partir du code de Huffman associé à la statistique des symboles 1 à 13 en remplaçant le symbole 1 par le symbole 0=plage de 1 si la longueur moyenne de cette plage s'avère être supérieure à 2 (voir paragraphe 5.1.3). Pour représenter chacune des p parties connexes du graphe de contours, il faut adresser un de ses points caractérisé par deux coordonnées de 8 bits. Si l'on décide de coder les positions de ces points relativement les uns par rapport aux autres, on peut admettre que 14 bits suffisent. On peut donc écrire:

$$C_c = 14p + \min(\text{Huff}(1,2,\dots,13), \text{Huff}(0,2,3,\dots,13) + \text{Huff}_0) \quad (6.2)$$

où $\text{Huff}(x,y,\dots)$ définit le coût calculé selon Huffman de la suite des symboles x,y,\dots dont on a estimé la distribution statistique et Huff_0 correspond au coût calculé selon Huffman à partir de la statistique des longueurs de plage de 1.

Le coût moyen b_c associé à un point de l'image interfrontière 4-connexe s'obtient en divisant C_c par le nombre de points frontières correspondant n_c . Sa valeur moyenne devrait se situer aux environs de 1.5 bit par point.

Pour ce qui est du coût associé à l'information texture, trois termes interviennent dans son évaluation. A partir du seuil T , on connaît le nombre de surfaces de reconstruction N_0, N_1, N_2 et N_3 de chaque degré de polynôme. Pour L surfaces de reconstruction, on estime alors le coût selon Huffman nécessaire à établir le degré du polynôme qui les caractérise. Il faut ensuite coder les coefficients liés à chacun de ces polynômes. Le dernier terme correspond aux adresses qu'il faut utiliser pour associer aux M domaines disjoints de l'image segmentée une des L surfaces de reconstruction. Le coût associé à l'information texture est donc donné par:

$$C_t = \text{Huff}(N_0, N_1, N_2, N_3) + \sum_{i=0}^3 4(i+1)(i+2)N_i + M \log_2 L \quad (6.3)$$

Si le nombre de surfaces de reconstruction est identique au nombre de domaines disjoints de la segmentation ($L=M$), on peut supprimer le dernier terme de l'expression (6.3).

Le coût associé à la transmission des faux contours à rehausser est donné par (voir section 5.3):

$$C_r = 12n_e \quad (6.4)$$

où n_e correspond au nombre de points constituant des extrémités de faux contours à transmettre.

A partir des relations (6.1) à (6.4), on trouve l'expression du facteur de compression. En effet, on peut écrire:

$$\rho = (256^2 \times 8) / C = 524288 / C \quad (6.5)$$

Maintenant que tous les aspects de présentation des résultats ont été définis, examinons donc ces résultats.

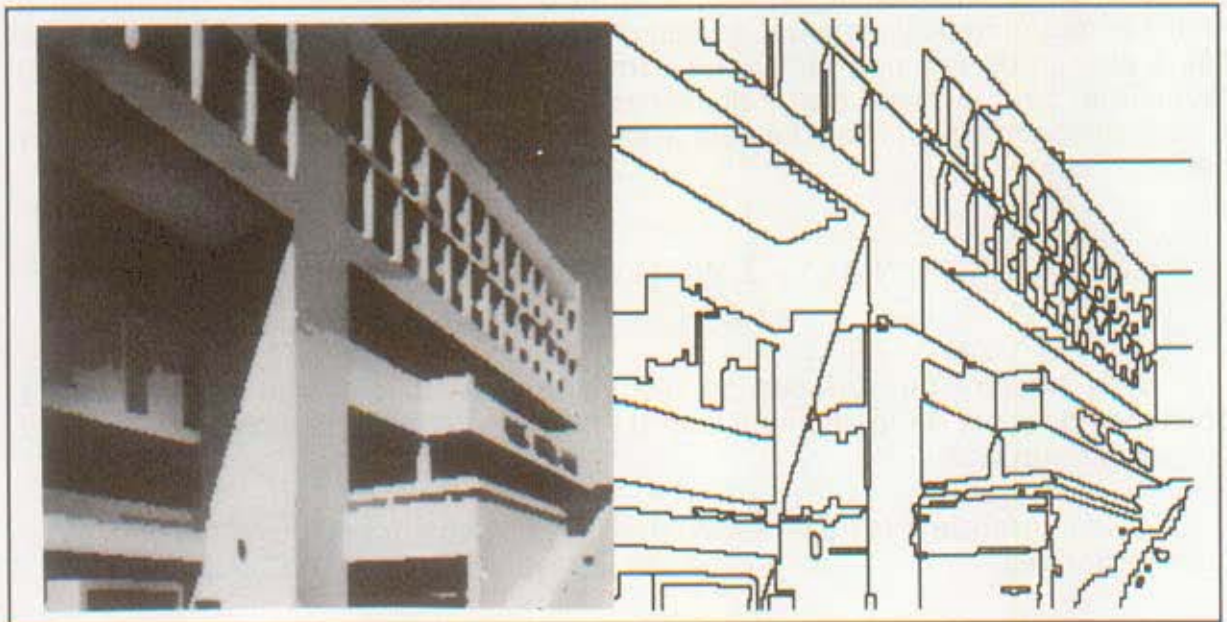
6.2 Résultats expérimentaux et discussion :

Pour chaque figure, la partie a correspond à l'image reconstruite après toute la procédure de segmentation, codage et rehaussement éventuel. La partie b constitue l'image interfrontière 4-connexe définissant la forme de la segmentation. Les paramètres non indiqués prennent leur valeur par défaut définie à la section précédente.

6.2.1 Segmentation adaptative par polynômes de degré 0:

Les premières dix figures correspondent aux résultats de la méthode de codage lorsqu'on utilise une approximation par constantes pour effectuer la segmentation adaptative dans son ensemble ($\alpha=0$).

Les figures 6.5 à 6.7 montrent les images du bâtiment reconstruites pour des segmentations formées de $N_r=99$, 49 et 24 régions, respectivement. Les facteurs de compression associés sont de 35 à 1, 52 à 1, et 80 à 1. Vu l'inexistence d'une composante aléatoire globale, on n'a pas rajouté de signal aléatoire à ces images.



a) Image reconstruite

$\alpha=0$ $N_r=99$
 $M=128$ $L=99$
 $N_0=5$ $N_1=9$
 $\text{Huff}(1,2,\dots,13) = 10049$
 $\text{Huff}(N_0,N_1,N_2,N_3)=113$

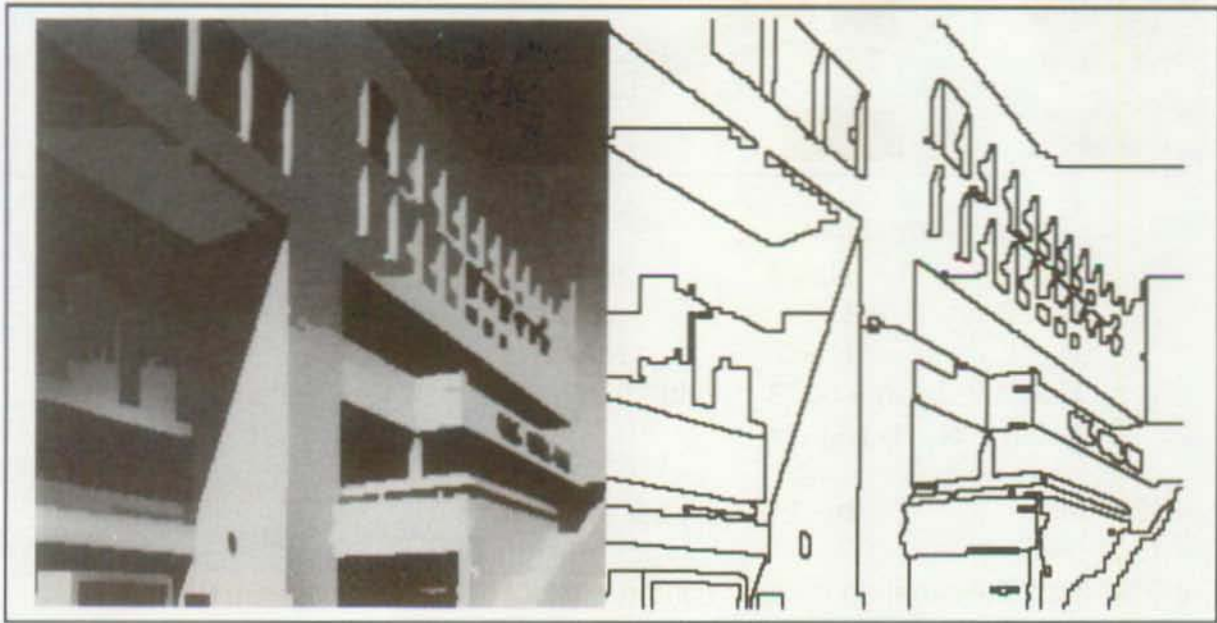
b) Image interfrontière 4-connexe.

$A=999$ $T=5^2$
 $p=15$ $n_c=7169$
 $N_2=85$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 7092$ $\text{Huff}_0 = 2438$

$C_c = 9740$ bits $b_c = 1.359$ bit par point contour.
 $C_t = 5345$ bits
 $C_r = 0$ bit (pas d'élimination de faux contours: $n_e = 0$) ; pas de composante aléatoire.
 $C = 15085$ bits $\rho = 34.75$

Figure 6.5: Image du bâtiment reconstruite après segmentation en 99 régions.

L'image obtenue à partir de la segmentation arrêtée à 99 régions est de bonne qualité. On distingue presque tous les détails de l'image originale et presque aucune distorsion au niveau des contours réels. En observant la forme de la segmentation, on peut remarquer qu'une zone uniforme comme le ciel du bâtiment est divisée en plusieurs parties. La non existence de faux contours dans cette partie de l'image reconstruite montre que ces parties peuvent parfaitement être associées sans distorsion si l'on poursuit le rassemblement avec des polynômes de degré plus élevé. Le facteur de compression est particulièrement élevé pour ce niveau de qualité. On peut s'en rendre compte en comparant cette image aux résultats de Kocher [27, p.206] pour la méthode de croissance de régions appliquée à l'image du bâtiment. On a triplé la réduction de redondance par rapport à l'image divisée de la figure 3.24c sans perte sensible de qualité.



a) Image reconstruite

$\alpha = 0$ $N_r = 49$
 $M = 65$ $L = 49$
 $N_0 = 2$ $N_1 = 5$
 $\text{Huff}(1, 2, \dots, 13) = 7597$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 56$

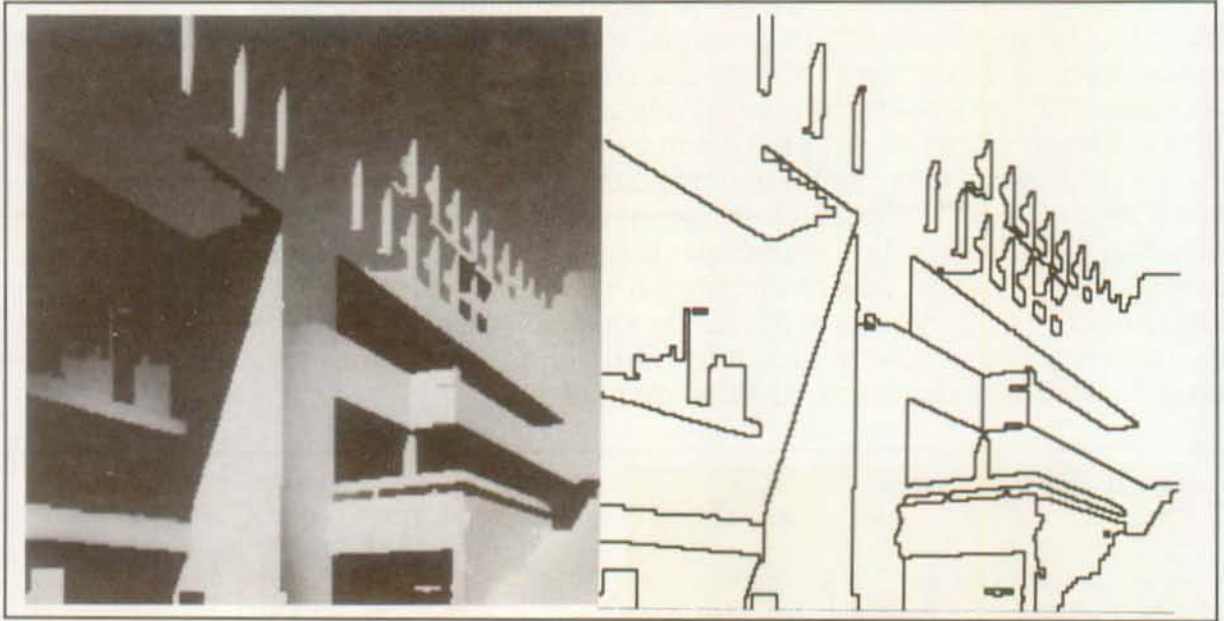
b) Image interfrontière 4-connexe.

$A = 999$ $T = 5^2$
 $p = 19$ $n_c = 5692$
 $N_2 = 42$ $N_3 = 0$
 $\text{Huff}(0, 2, 3, \dots, 13) = 5235$ $\text{Huff}_0 = 1897$

$C_c = 7398$ bits $b_c = 1.300$ bit par point contour.
 $C_t = 2598$ bits
 $C_r = 0$ bit (pas d'élimination de faux contours: $n_e = 0$) ; pas de composante aléatoire.
 $C = 9996$ bits $\rho = 52.45$

Figure 6.6: Image du bâtiment reconstruite après segmentation en 49 régions.

La compression augmente (de 35 à 1 à 52 à 1) en réduisant le nombre de régions de 99 à 49 régions. On n'est cependant plus à même de garantir à ce niveau la correspondance de toutes les frontières de régions avec des contours réels. Ainsi l'arête supérieure du bâtiment a été associée au ciel. Il est encore possible de comprendre le contenu de l'image à ce niveau. Par rapport à l'image de la figure 6.5a, on peut remarquer aussi que certaines fenêtres ont disparu.



a) Image reconstruite

$\alpha=0$ $N_r=24$
 $M=33$ $L=24$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 5273$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 24$

b) Image interfrontière 4-connexe.

$A=999$ $T=5^2$
 $p=15$ $n_c=3901$
 $N_2=23$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 3748$ $\text{Huff}_0 = 1310$

$C_c = 5268$ bits

$b_c = 1.350$ bit par point contour.

$C_t = 1301$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

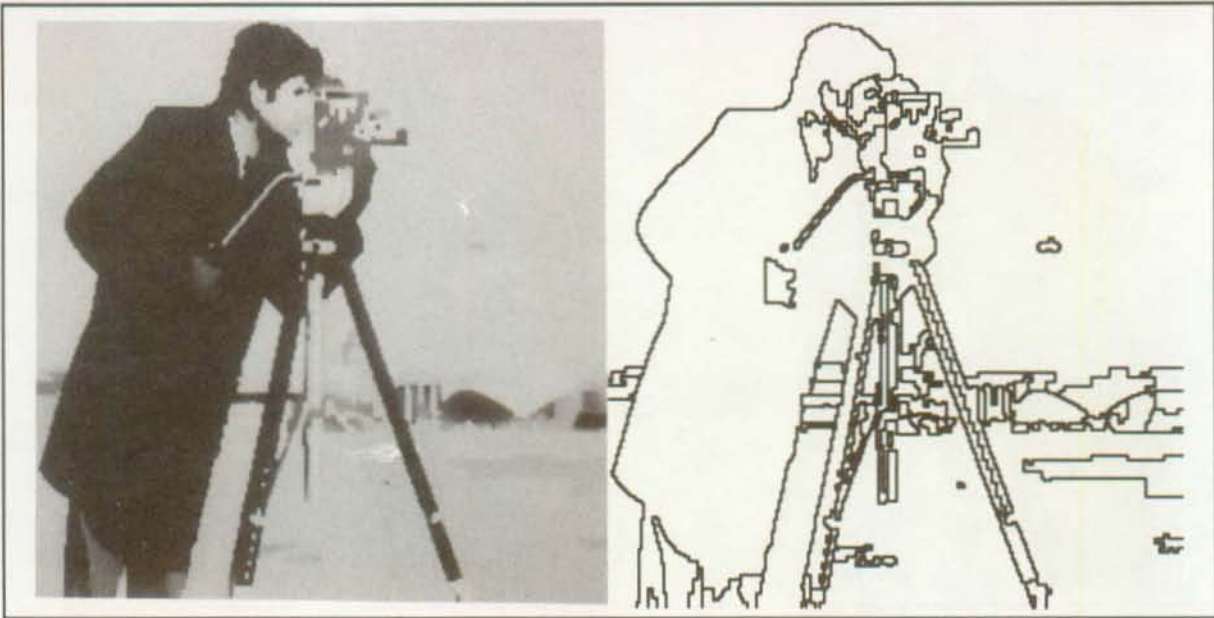
$C = 6569$ bits

$\rho = 79.81$

Figure 6.7: Image du bâtiment reconstruite après segmentation en 24 régions.

Si la compression d'information atteint des niveaux très élevés pour les résultats de la figure 6.7, l'image est ici de très faible qualité. Les distorsions qui existaient au niveau de l'image de la figure 6.6 se sont accentuées (destruction de la partie supérieure du bâtiment). Seules sont préservées les régions de grande taille très contrastées par rapport à leurs voisines.

Les figures 6.8 à 6.11 montrent les images du caméraman reconstruites pour des segmentations formées de $N_r=99$, 49, 24 et 9 régions, respectivement. Les facteurs de compression associés sont de 41 à 1, 55 à 1, 79 à 1 et 121 à 1. On n'a pas rajouté de composante aléatoire aux images reconstruites.



a) Image reconstruite

$\alpha=0$ $N_r=99$
 $M=109$ $L=79$
 $N_0=4$ $N_1=6$
 $\text{Huff}(1,2,\dots,13) = 8273$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 89$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=16$ $n_c=5131$
 $N_2=69$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6607$ $\text{Huff}_0 = 1663$

$C_c = 8494$ bits $b_c = 1.655$ bit par point contour.

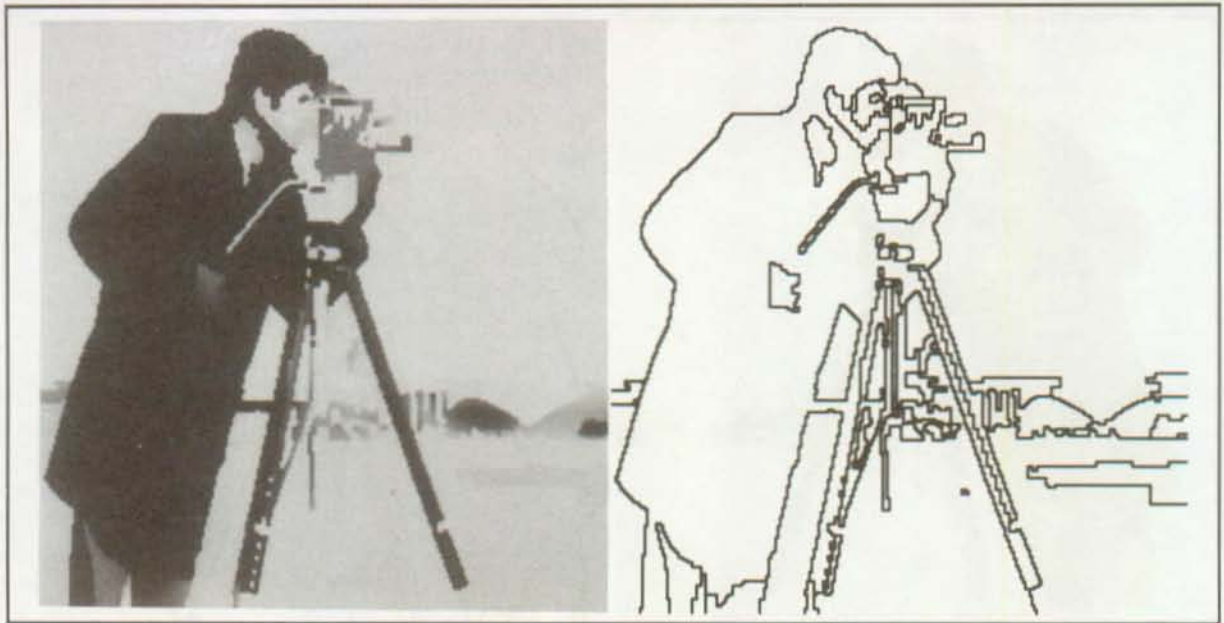
$C_t = 4340$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

$C = 12834$ bits $\rho = 40.85$

Figure 6.8: Image du caméraman reconstruite après segmentation en 99 régions.

L'image du caméraman obtenue à partir de la segmentation arrêtée à 99 régions est de bonne qualité. On peut faire le même genre de remarque que pour le résultat de la segmentation du bâtiment en 99 régions (figure 6.5). Un grand niveau de détail reste présent au niveau de la caméra même après élimination des liens de régions. Presque toutes les régions obtenues sont des parties d'objet de l'image. Le facteur de compression est particulièrement élevé pour ce niveau de qualité. On peut s'en rendre compte en comparant cette image aux résultats de Kocher [27, p.204] pour la méthode de croissance de régions appliquée à l'image du caméraman. On a triplé la réduction de redondance par rapport à l'image divisée de la figure 3.25c sans perte sensible de qualité. Le seul désagrément sensible que l'on trouve dans cette image correspond à la mauvaise reproduction de la texture du gazon. Ceci pourra être amélioré par adjonction d'une composante aléatoire de signal relativement aux régions de gazon (voir paragraphe 6.4.2).



a) Image reconstruite

 $\alpha=0$ $N_r=49$ $M=69$ $L=47$ $N_0=1$ $N_1=3$

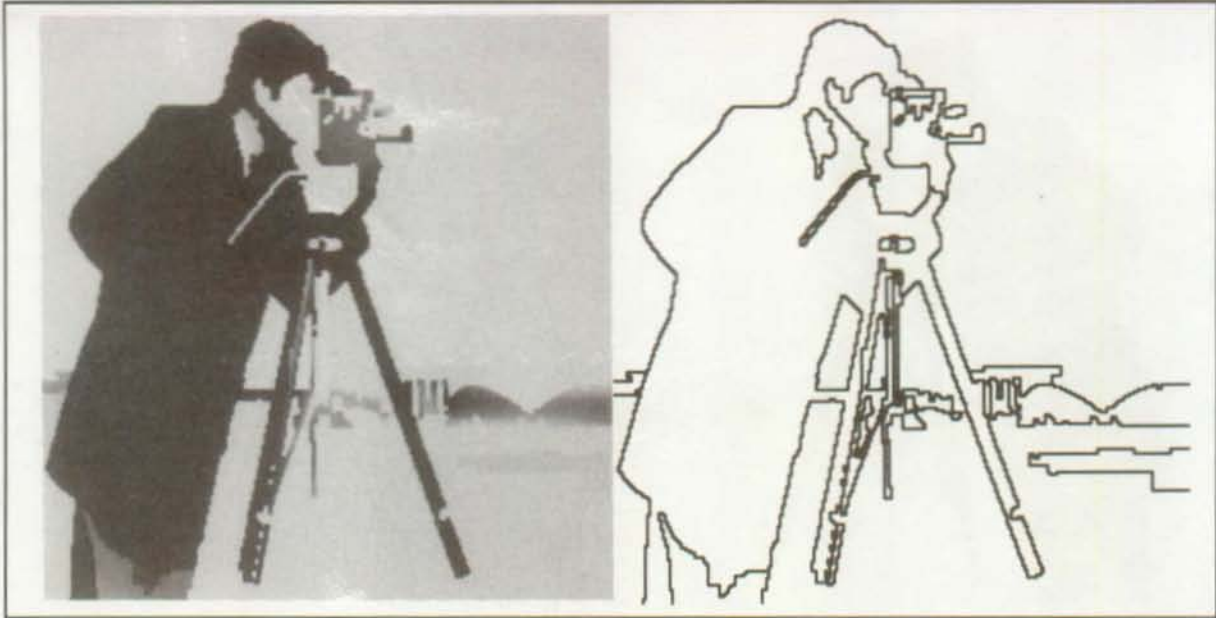
Huff(1,2,...,13) = 6740

Huff(N_0, N_1, N_2, N_3)=51

b) Image interfrontière 4-connexe.

 $A=499$ $T=5^2$ $p=9$ $n_c=4413$ $N_2=43$ $N_3=0$ Huff(0,2,3,...,13) = 5292 Huff₀ = 1460 $C_c = 6866$ bits $b_c = 1.556$ bit par point contour. $C_t = 2609$ bits $C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire. $C = 9475$ bits $\rho = 55.33$ Figure 6.9: Image du caméraman reconstruite après segmentation en 49 régions.

La compression augmente de 33% (de 42 à 1 à 56 à 1) en réduisant le nombre de régions de 99 à 49 régions. Contrairement à l'image du bâtiment, cette réduction n'est pas accompagnée d'une perte sensible de qualité. Des régions de petite taille ont été associées à de plus grands domaines; sur le plan sémantique, ces régions ne correspondent pas à détails essentiels, d'où le niveau de qualité comparable des images des figures 6.8 et 6.9.



a) Image reconstruite

$\alpha=0$ $N_r=24$
 $M=38$ $L=23$
 $N_0=0$ $N_1=3$
 $\text{Huff}(1,2,\dots,13) = 5353$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 23$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=8$ $n_c=3644$
 $N_2=20$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4096$ $\text{Huff}_0 = 1178$

$C_c = 5386$ bits

$b_c = 1.478$ bit par point contour.

$C_l = 1244$ bits

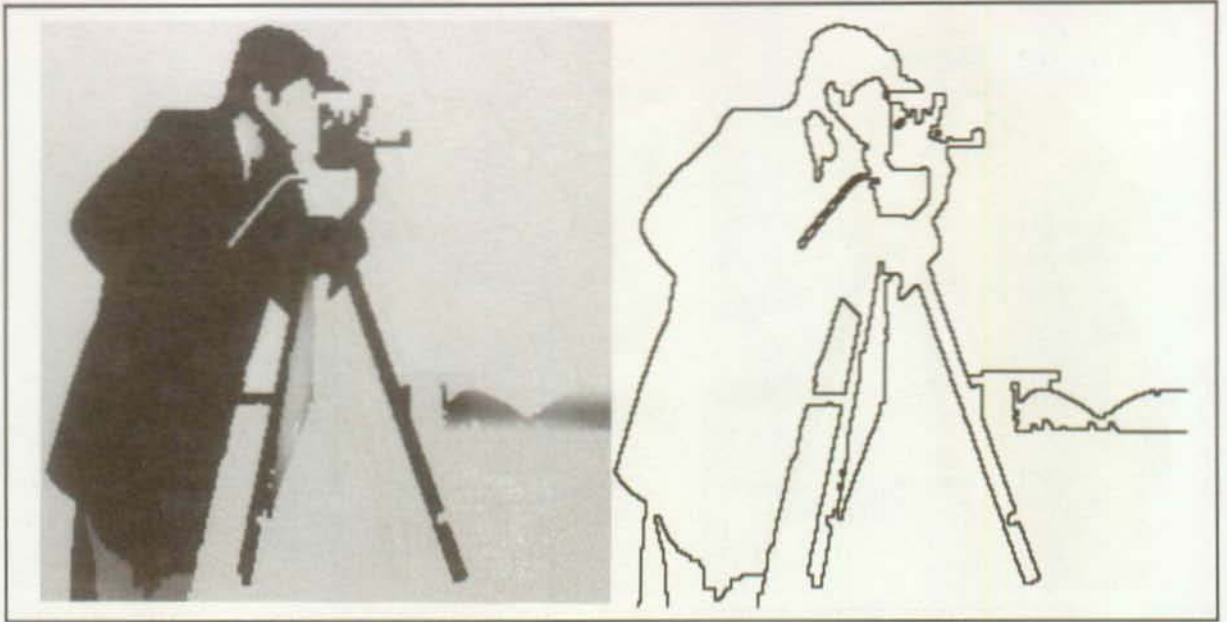
$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

$C = 6630$ bits

$\rho = 79.07$

Figure 6.10: Image du caméraman reconstruite après segmentation en 24 régions.

La compression d'information atteint des niveaux très élevés pour l'image de la figure 6.10. L'image est de qualité moyenne. On peut parfaitement distinguer la caméra alors que certains détails ont été perdus au niveau de l'oeil du caméraman et de la séparation entre les jambes de son pantalon. La différence de qualité est cependant appréciable si l'on compare l'image de la figure 6.10a à celle du bâtiment formée du même nombre de régions. Ainsi, s'il existe de grands espaces réguliers nettement contrastés, on peut réduire sensiblement le nombre de régions de l'image tout en conservant une grande partie de son contenu sémantique. Cette hypothèse est confirmée par le résultat de la figure 6.11 où le nombre de régions a été réduit à 9 régions. Dans cette dernière image, on observe uniquement comme distorsion sensible l'artéfact de la partie gauche du trépied.



a) Image reconstruite

$\alpha=0$ $N_r=9$
 $M=13$ $L=9$
 $N_0=0$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 3766$
 $\text{Huff}(N_0,N_1,N_2,N_3)=0$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=5$ $n_c=2711$
 $N_2=9$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 2950$ $\text{Huff}_0 = 889$

$C_c = 3836$ bits

$b_c = 1.415$ bit par point contour.

$C_1 = 484$ bits

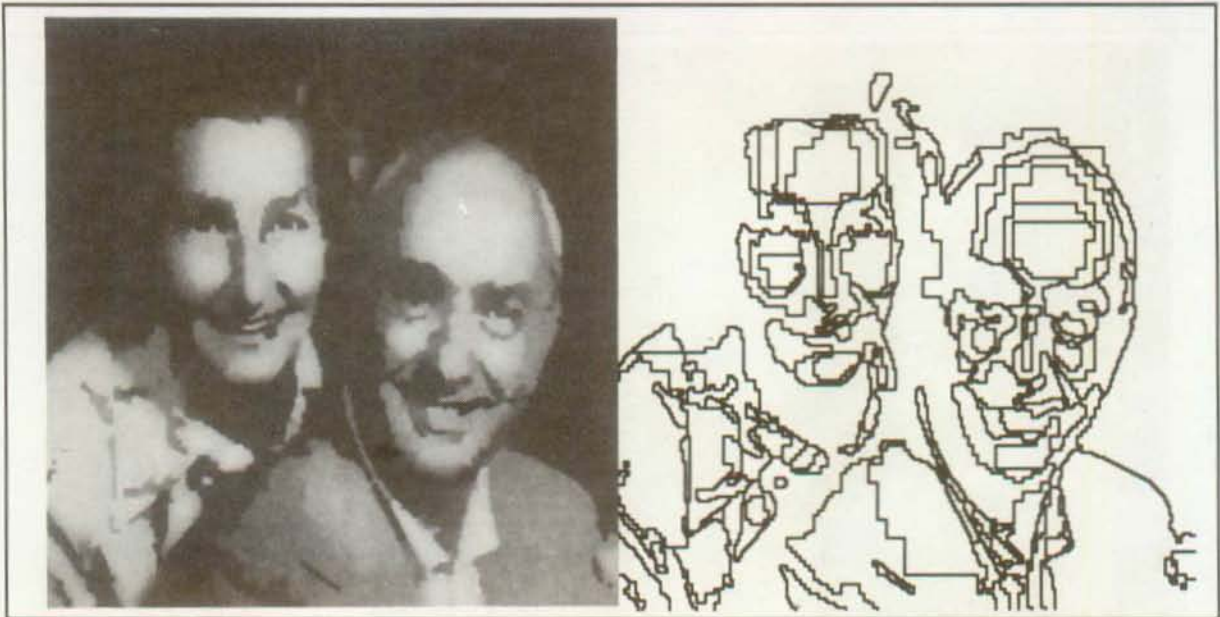
$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

$C = 4320$ bits

$p = 121.36$

Figure 6.11: Image du caméraman reconstruite après segmentation en 9 régions.

Les figures 6.12 à 6.14 montrent les images du couple reconstruites pour des segmentations formées de $N_r=99$, 49 et 24, respectivement. Les facteurs de compression associés sont de 29 à 1, 43 à 1 et 66 à 1. Comme les espaces réguliers correspondent à des surfaces polynomiales du second degré, on a rehaussé la qualité de l'image reconstruite par lissage des faux contours. Pour donner un aspect plus naturel, une composante aléatoire de signal (bruit blanc gaussien d'écart type $\sigma=10$ filtré passe-bas par un moyennneur 3×3) est rajoutée sur toute l'image. Ces paramètres sont estimés d'une façon globale à partir du signal différence entre l'image reconstruite par des fonctions polynomiales et le signal original. La variance du bruit correspond à la puissance du signal de différence en admettant que ce dernier ait une valeur moyenne nulle. Le bruit blanc généré est filtré passe-bas de manière à tenir compte de la fréquence maximum des variations que l'on désire reproduire. Les estimations sont faussées par le fait que certains objets ou parties d'objets de l'image originale ont disparu. La valeur des paramètres utilisés sera justifiée à la section 6.4.



a) Image reconstruite

$\alpha=0$ $N_r=99$
 $M=139$ $L=99$
 $N_0=1$ $N_1=4$
 $\text{Huff}(1,2,\dots,13) = 11320$
 $\text{Huff}(N_0,N_1,N_2,N_3)=104$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=13$ $n_c=7528$
 $N_2=94$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 9307$ $\text{Huff}_0 = 2239$

$C_c = 11502$ bits

$b_c = 1.528$ bit par point contour.

$C_l = 5693$ bits

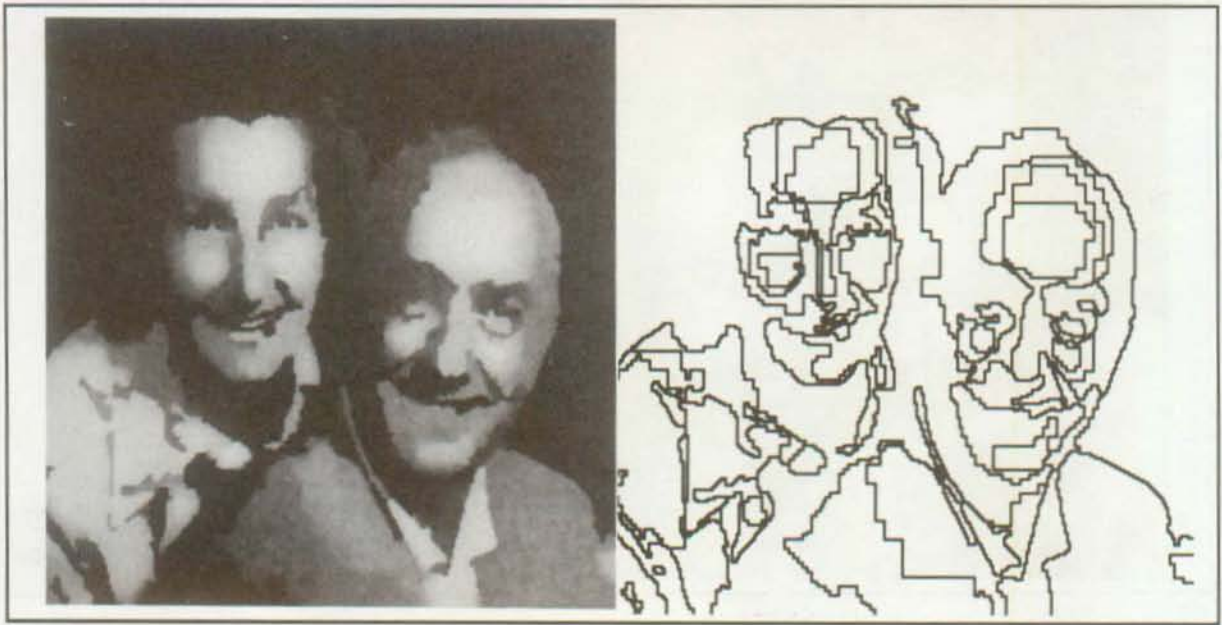
$C_r = 684$ bits (élimination des faux contours dépassant $d=20:n_e=57$); adjonction d'une composante aléatoire.

$C = 17879$ bits

$\rho = 29.32$

Figure 6.12: Image du couple reconstruite après segmentation en 99 régions.

Comme pour les deux autres images traitées, la qualité de l'image du couple reconstruite est élevée pour une segmentation formée de 99 régions. Cette image se distingue des précédentes par le fait qu'elle requiert un nombre plus élevé de polynômes de degrés élevés ($N_2=94$ au lieu de 69 pour l'image du caméraman et de 85 pour l'image du bâtiment). Ceci s'explique par le genre de variation caractérisant des zones uniformes de l'image du couple. Vu la forme relativement complexe des contours de l'image, et donc des frontières de régions qui leur sont associées, le coût lié à l'information contour apparaît également plus élevé ($C_c=11502$ au lieu de 8494 pour l'image du caméraman et de 9740 pour l'image du bâtiment). Il en résulte un facteur de compression légèrement plus faible pour cette image. La forme compliquée de la segmentation montre comment il est difficile de suivre des surfaces à variations complexes en effectuant la segmentation avec des constantes. Il en résulte une subdivision excessive des fronts des personnages du couple. Ceci se fait au détriment de certains détails tels la perte des cheveux de la femme. On peut comparer l'image de la figure 6.12 aux résultats de Kocher [27, p.208] pour la méthode de croissance de régions appliquée à l'image du couple. On a doublé la réduction de redondance par rapport à l'image divisée de la figure 3.26c sans perte sensible de qualité.



a) Image reconstruite

$\alpha=0$ $N_r=49$
 $M=67$ $L=49$
 $N_0=1$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 8621$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 51$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=9$ $n_c=6034$
 $N_2=47$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6824$ $\text{Huff}_0 = 2106$

$C_c = 8747$ bits

$b_c = 1.450$ bit par point contour.

$C_t = 2741$ bits

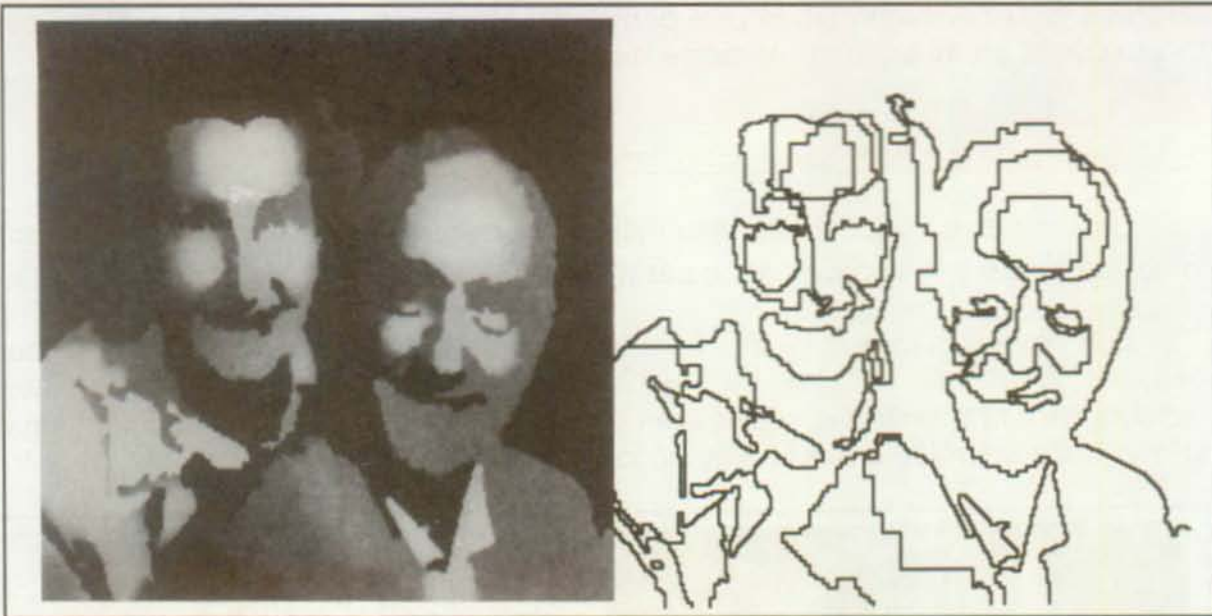
$C_r = 684$ bits (élimination des faux contours dépassant $d=20$: $n_e=57$); adjonction d'une composante aléatoire.

$C = 12172$ bits

$\rho = 43.07$

Figure 6.13: Image du couple reconstruite après segmentation en 49 régions.

Pour la figure 6.13, la compression augmente de 48% (de 29 à 1 à 43 à 1) en réduisant le nombre de régions de 99 à 49 régions. Contrairement à l'image du bâtiment, cette réduction n'est pas accompagnée d'une perte importante de qualité. Cependant, si les contours importants n'ont pas été détruits, on a perdu en finesse de reproduction de certaines parties de l'image. Ainsi, l'oeil droit de l'homme est un peu déformé de même que la partie droite de son front. Il en va de même des détails des plis de sa veste et de son menton.



a) Image reconstruite

$\alpha=0$ $N_r=24$
 $M=33$ $L=24$
 $N_0=0$ $N_1=2$
 $\text{Huff}(1,2,\dots,13) = 6337$
 $\text{Huff}(N_0,N_1,N_2,N_3)=24$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=8$ $n_c=4776$
 $N_2=22$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4935$ $\text{Huff}_0 = 1707$

$C_c = 6449$ bits

$b_c = 1.350$ bit par point contour.

$C_t = 1293$ bits

$C_r = 228$ bits (élimination des faux contours dépassant $d=20:n_e=19$); adjonction d'une composante aléatoire.

$C = 7970$ bits

$\rho = 65.78$

Figure 6.14: Image du couple reconstruite après segmentation en 24 régions.

La compression d'information atteint des niveaux très élevés (moins de 8000 bits suffisent pour reconstruire l'image de la figure 6.14). L'image a perdu en qualité cependant moins à cause de la perte de contours réels que par la faiblesse de reproduction de certains détails importants liés aux visages des êtres humains. En effet, l'uniformité excessive de la partie gauche du visage de l'homme est très désagréable. La perte des dents des personnages ôte aussi une grande partie du réalisme des visages. Si l'on est toujours en mesure de distinguer deux personnes sur l'image de la figure 6.14a, il devient plus difficile de les reconnaître. On a probablement éliminé des caractéristiques importantes quant à la représentation de ces visages dans notre cerveau humain.

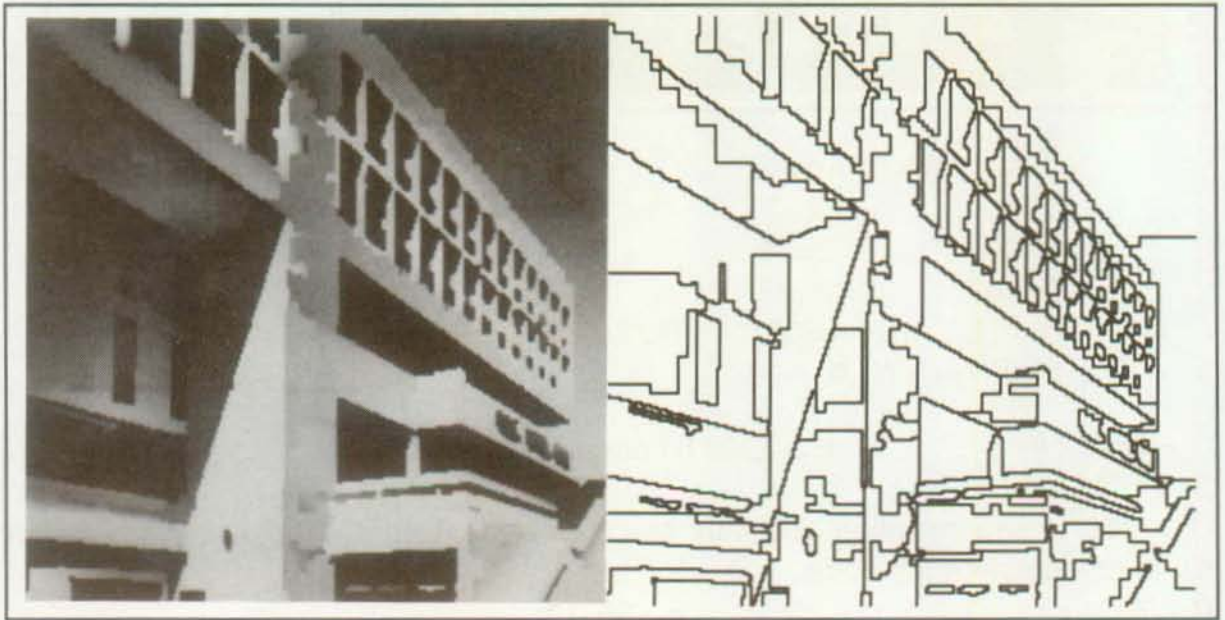
De tous ces résultats, il ressort que les frontières des régions obtenues suivent presque idéalement les contours réels des objets préservés par la segmentation. L'information n'apparaît distordue que par la disparition de régions sémantiquement significatives (arête supérieure du bâtiment, oeil du caméraman, dents des personnages). A cause du choix d'effectuer les segmentations par des polynômes de degré 0, il n'est pratiquement pas possible de suivre des surfaces à variations complexes bien que lentes. Des régions artificielles (subdivision du ciel, front des personnages) sont ainsi créées et ne disparaîtront lors du rassemblement

qu'après la perte d'information plus pertinente. Voyons donc comment pallier à cet inconvénient en augmentant le degré du polynôme d'approximation.

6.2.2 Segmentation adaptative par polynômes de degré 1:

Les figures 6.15 à 6.23 correspondent aux résultats de la méthode de codage lorsqu'on utilise une approximation par plans pour effectuer la segmentation ($\alpha=1$).

Les figures 6.15 à 6.17 montrent les images du bâtiment reconstruites pour des segmentations formées de $N_r=99$, 49 et 24 régions, respectivement. Les facteurs de compression associés sont de 31 à 1, 46 à 1, et 72 à 1. Aucun signal aléatoire n'a été rajouté aux images reconstruites.



a) Image reconstruite

$\alpha=1$ $N_r=99$
 $M=122$ $L=98$
 $N_0=2$ $N_1=9$
 $\text{Huff}(1,2,\dots,13) = 11654$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 109$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=17$ $n_c=8093$
 $N_2=87$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 8440$ $\text{Huff}_0 = 2749$

$C_c = 11427$ bits

$b_c = 1.412$ bit par point contour.

$C_t = 5371$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

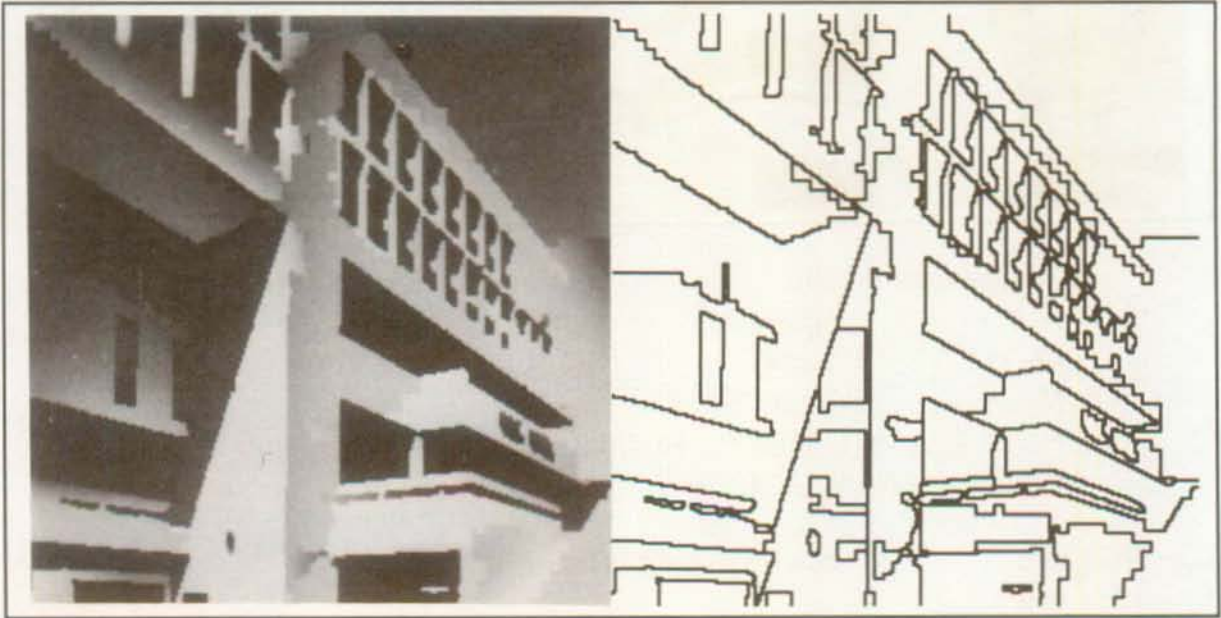
$C = 16798$ bits

$p = 31.21$

Figure 6.15: Image du bâtiment reconstruite après segmentation en 99 régions.

Si l'on compare l'image de la figure 6.15a à l'image de la figure 6.5a, on observe certaines imprécisions au niveau des contours réels de l'image reconstruite (voir arête supérieure droite du bâtiment). Les fenêtres du bâtiment sont néanmoins mieux reproduites dans ce cas ainsi que la partie gauche de l'image. Les imprécisions au niveau des contours sont dues à une mauvaise

association de petites régions situées dans le voisinage de ces défauts au cours du processus de rassemblement. Un plus grand nombre de détails a pu être préservé à cause de l'augmentation du degré du polynôme qui permet d'associer des parties d'espaces réguliers plus complexes que de simples constantes. Ainsi, la zone homogène correspondant au ciel du bâtiment n'est plus séparée qu'en deux régions au lieu de cinq. La part des frontières artificielles semble cependant plus élevée pour cette image que pour celle de la figure 6.5a. Il en résulte un plus grand nombre de points contours (n_c) et par conséquent une augmentation du coût correspondant lié à l'information de contour (C_c). Le facteur de compression ne peut donc qu'être légèrement moins élevé.



a) Image reconstruite

$\alpha=1$ $N_r=49$
 $M=65$ $L=49$
 $N_0=2$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 8110$
 $\text{Huff}(N_0,N_1,N_2,N_3)=52$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=14$ $n_c=6354$
 $N_2=46$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6327$ $\text{Huff}_0 = 2205$

$C_c = 8728$ bits

$b_c = 1.374$ bit par point contour.

$C_t = 2690$ bits

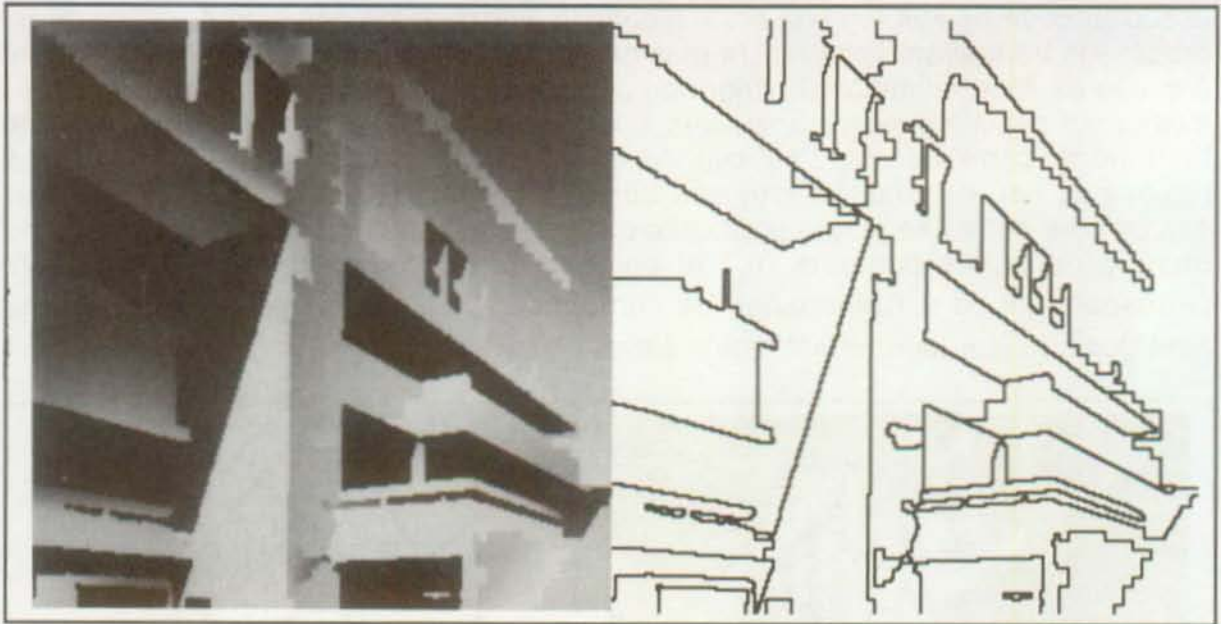
$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

$C = 11418$ bits

$\rho = 45.91$

Figure 6.16: Image du bâtiment reconstruite après segmentation en 49 régions.

Si la compression est plus faible dans ce cas de figure (6.16) par rapport au résultat de la figure 6.6, la qualité de l'image en est nettement plus élevée. Certes, on a perdu quelques fenêtres et déformé une partie du ciel mais la structure du bâtiment est préservée. Le même genre d'imperfections de la figure 6.15 apparaît au niveau de la reproduction des contours réels de l'image.



a) Image reconstruite

$\alpha=1$ $N_r=24$
 $M=31$ $L=24$
 $N_0=2$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 5988$
 $\text{Huff}(N_0,N_1,N_2,N_3)=24$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=10$ $n_c=4605$
 $N_2=22$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4309$ $\text{Huff}_0 = 1612$

$C_c = 6061$ bits

$b_c = 1.316$ bit par point contour.

$C_t = 1251$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

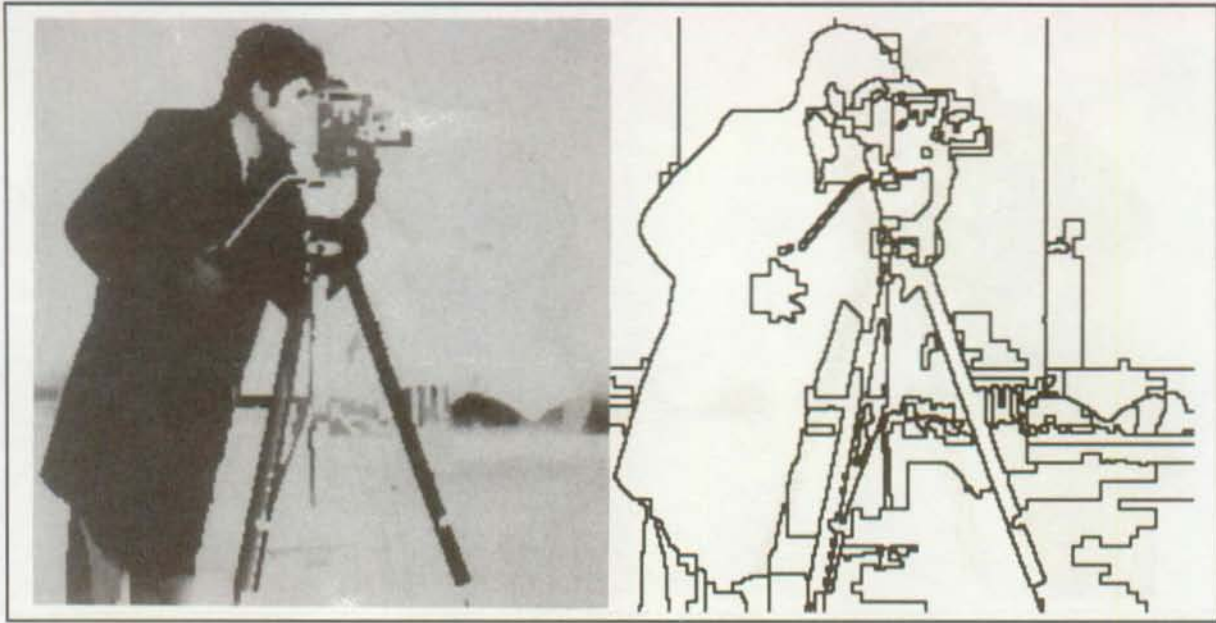
$C = 7312$ bits

$\rho = 71.70$

Figure 6.17: Image du bâtiment reconstruite après segmentation en 24 régions.

La compression est très élevée pour l'image de la figure 6.17a mais sa qualité est inacceptable. Si la partie supérieure du bâtiment est préservée dans l'image de la figure 6.17a, presque toutes les fenêtres ont disparu.

Les figures 6.18 à 6.20 montrent les images du caméraman reconstruites pour des segmentations formées de $N_r=99$, 49 et 24 régions, respectivement. Les facteurs de compression associés sont de 40 à 1, 53 à 1 et 81 à 1. Ils sont donc très proches des valeurs obtenues pour les figures 6.8 à 6.11. Voyons les différences correspondantes de qualité et de forme de segmentation. Pour donner une impression plus naturelle aux images reconstruites par des fonctions polynomiales, on a choisi d'ajouter la même composante aléatoire que celle utilisée pour les figures liées à l'image du couple (voir figures 6.12 à 6.14).



a) Image reconstruite

$\alpha=1$ $N_r=99$
 $M=112$ $L=81$
 $N_0=3$ $N_1=8$
 $\text{Huff}(1,2,\dots,13) = 8273$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 92$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=8$ $n_c=5710$
 $N_2=70$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6607$ $\text{Huff}_0 = 1663$

$C_c = 8704$ bits

$b_c = 1.524$ bit par point contour.

$C_t = 4452$ bits

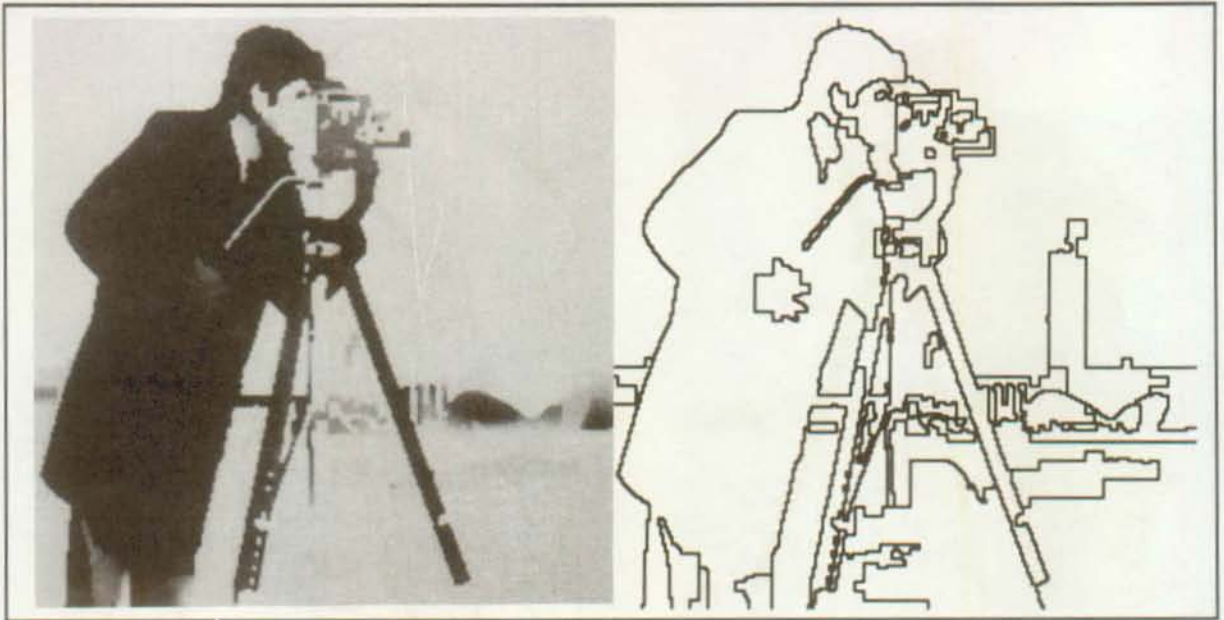
$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); adjonction d'une composante aléatoire.

$C = 13156$ bits

$\rho = 39.85$

Figure 6.18: Image du caméraman reconstruite après segmentation en 99 régions.

L'image du caméraman obtenue à partir de la segmentation arrêtée à 99 régions est de bonne qualité. Par rapport à l'image de la figure 6.8, on peut remarquer qu'un plus grand nombre de détails ont été préservés (bâtiment allongé se détachant sur le fond de l'image). Ceci a pu être de nouveau réalisé grâce à l'augmentation du degré du polynôme. Comme l'image est plus contrastée, les artéfacts d'imprécision au niveau des frontières de régions sont moins apparants que dans l'image du bâtiment.



a) Image reconstruite

$\alpha=1$ $N_r=49$
 $M=69$ $L=47$
 $N_0=0$ $N_1=3$
 $\text{Huff}(1,2,\dots,13) = 7345$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 47$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=7$ $n_c=4762$
 $N_2=44$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 5570$ $\text{Huff}_0 = 1547$

$C_c = 7215$ bits

$b_c = 1.515$ bit par point contour.

$C_t = 2645$ bits

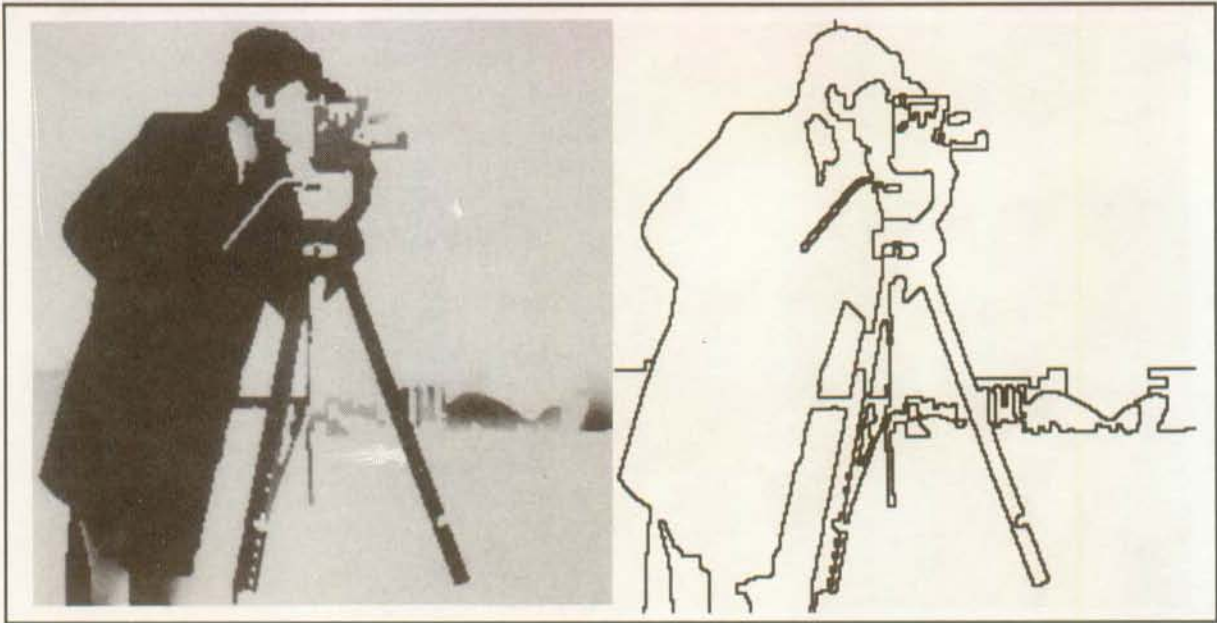
$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); adjonction d'une composante aléatoire.

$C = 9860$ bits

$\rho = 53.17$

Figure 6.19: Image du caméraman reconstruite après segmentation en 49 régions.

Le même genre de remarques effectuées pour la figure 6.9 s'appliquent à la figure 6.19. On observe une plus grande précision au niveau de l'image segmentée pour l'image ci-dessus qui explique la légère différence en compression.



a) Image reconstruite

$\alpha=1$ $N_r=24$
 $M=38$ $L=23$
 $N_0=0$ $N_1=4$
 $\text{Huff}(1,2,\dots,13) = 5159$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 23$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=7$ $n_c=3439$
 $N_2=19$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4025$ $\text{Huff}_0 = 1136$

$C_c = 5257$ bits

$b_c = 1.529$ bit par point contour.

$C_t = 1221$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

$C = 6478$ bits

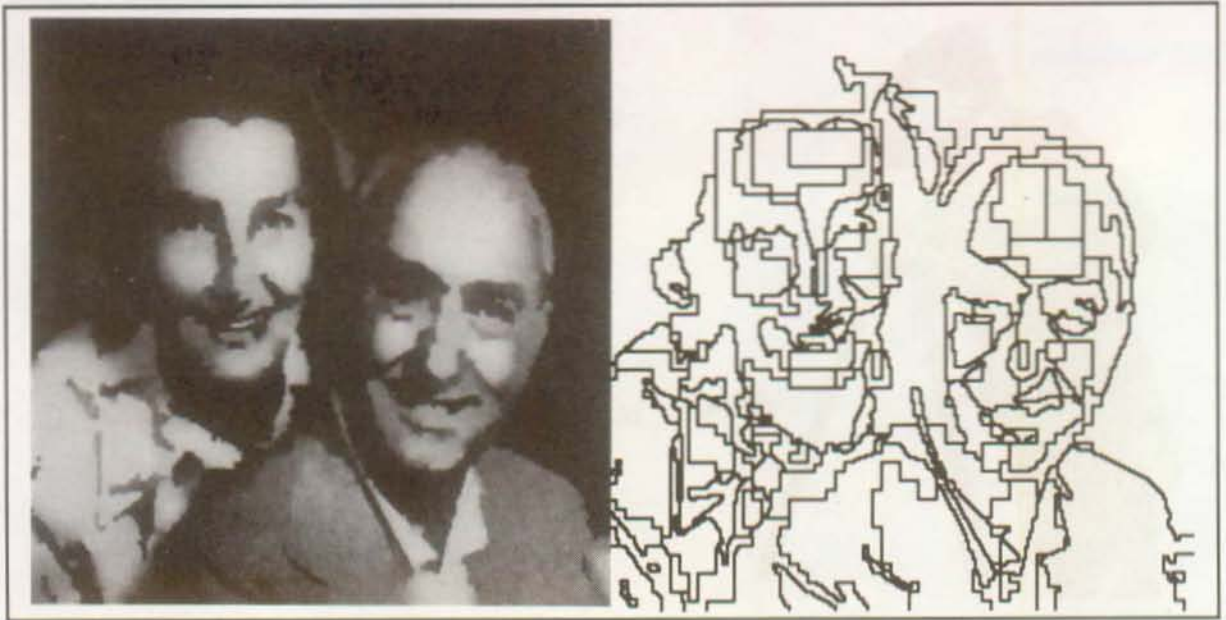
$\rho = 80.93$

Figure 6.20: Image du caméraman reconstruite après segmentation en 24 régions.

La compression d'information est très élevée pour l'image de la figure 6.20a. Les images des figures 6.10 et 6.20 sont parfaitement comparables.

En ce qui concerne l'adjonction d'une composante aléatoire, on peut observer une légère granulosité des régions à plages uniformes qui redonne un peu de réalisme aux images reconstruites.

Les figures 6.21 à 6.23 montrent les images du couple reconstruites pour des segmentations formées de $N_r=99$, 49 et 24, respectivement. Les facteurs de compression associés sont de 29 à 1, 44 à 1 et 61 à 1. Comme les espaces réguliers correspondent à des surfaces polynomiales du second degré, on a rehaussé la qualité de l'image reconstruite par lissage des faux contours. Pour donner un aspect plus naturel, la même composante aléatoire de signal que celle des figures 6.12 à 6.14 a été ajoutée.



a) Image reconstruite

$\alpha=1$ $N_r=99$
 $M=129$ $L=98$
 $N_0=1$ $N_1=7$
 $\text{Huff}(1,2,\dots,13) = 11255$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 114$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=9$ $n_c=7447$
 $N_2=90$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 8720$ $\text{Huff}_0 = 2569$

$C_c = 11415$ bits

$b_c = 1.533$ bit par point contour.

$C_1 = 5505$ bits

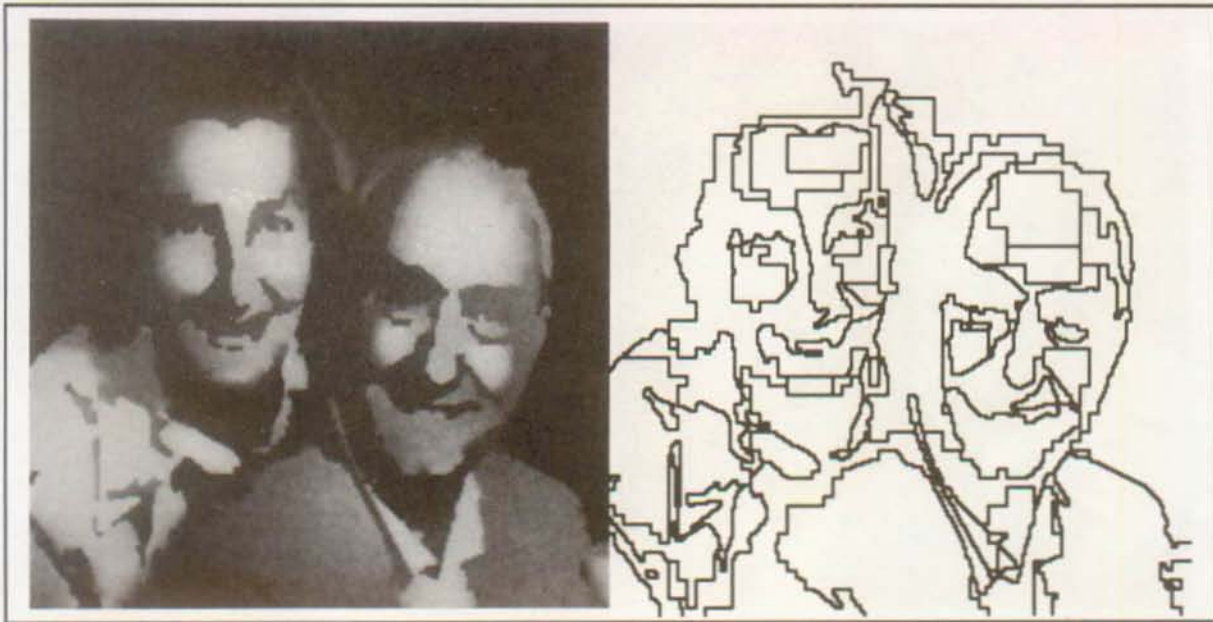
$C_r = 864$ bits (élimination des faux contours dépassant $d=20:n_e=72$); adjonction d'une composante aléatoire.

$C = 17784$ bits

$\rho = 29.48$

Figure 6.21: Image du couple reconstruite après segmentation en 99 régions.

Comme pour le bâtiment et le caméraman, la qualité de l'image du couple reconstruite est élevée pour une segmentation formée de 99 régions. L'augmentation du degré du polynôme entraîne comme dans les deux cas précédents l'apparition de détails supplémentaires au niveau de la segmentation (par exemple cheveux du côté droit du visage de la femme. Certaines imprécisions existent dans la reconstitution des contours réels de l'image de la figure 6.21 (par exemple nez de la femme) mais elles ont un effet moins désagréable pour cette image que pour l'image du bâtiment.



a) Image reconstruite

$\alpha=1$ $N_r=49$
 $M=66$ $L=49$
 $N_0=1$ $N_1=2$
 $\text{Huff}(1,2,\dots,13) = 8553$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 52$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=9$ $n_c=6015$
 $N_2=46$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6514$ $\text{Huff}_0 = 2122$

$C_c = 8679$ bits

$b_c = 1.443$ bit par point contour.

$C_t = 2712$ bits

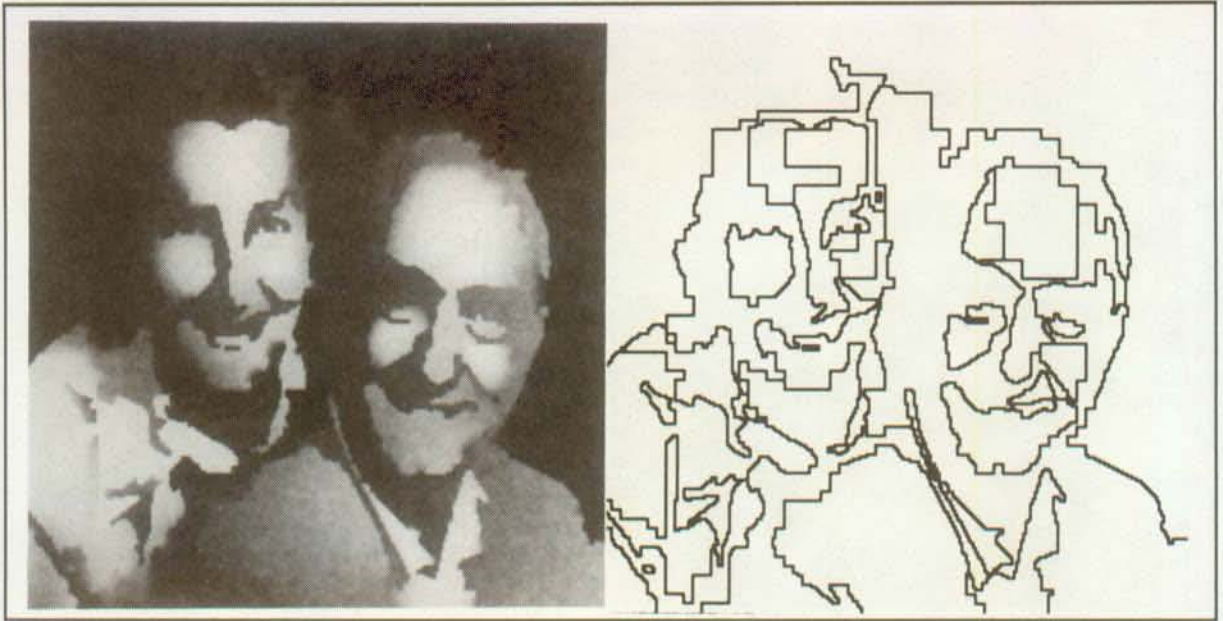
$C_r = 576$ bits (élimination des faux contours dépassant $d=20$: $n_e=48$); adjonction d'une composante aléatoire.

$C = 11967$ bits

$\rho = 43.81$

Figure 6.22: Image du couple reconstruite après segmentation en 49 régions.

Pour la figure 6.22, la compression augmente de 52% (de 29 à 1 à 44 à 1) en réduisant le nombre de régions de 99 à 49 régions. Contrairement à l'image du bâtiment, cette réduction n'est pas accompagnée d'une perte importante de qualité. Par rapport à l'image de la figure 6.13a on a moins perdu en finesse de reproduction de certaines parties de l'image (voir oeil droit de l'homme).



a) Image reconstruite

$\alpha=1$ $N_r=24$
 $M=33$ $L=24$
 $N_0=0$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 6828$
 $\text{Huff}(N_0, N_1, N_2, N_3)=24$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=8$ $n_c=5044$
 $N_2=23$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 5123$ $\text{Huff}_0 = 1779$

$C_c = 6940$ bits

$b_c = 1.376$ bit par point contour.

$C_t = 1301$ bits

$C_r = 324$ bits (élimination des faux contours dépassant $d=20:n_e=27$); adjonction d'une composante aléatoire.

$C = 8565$ bits

$\rho = 61.21$

Figure 6.23: Image du couple reconstruite après segmentation en 24 régions.

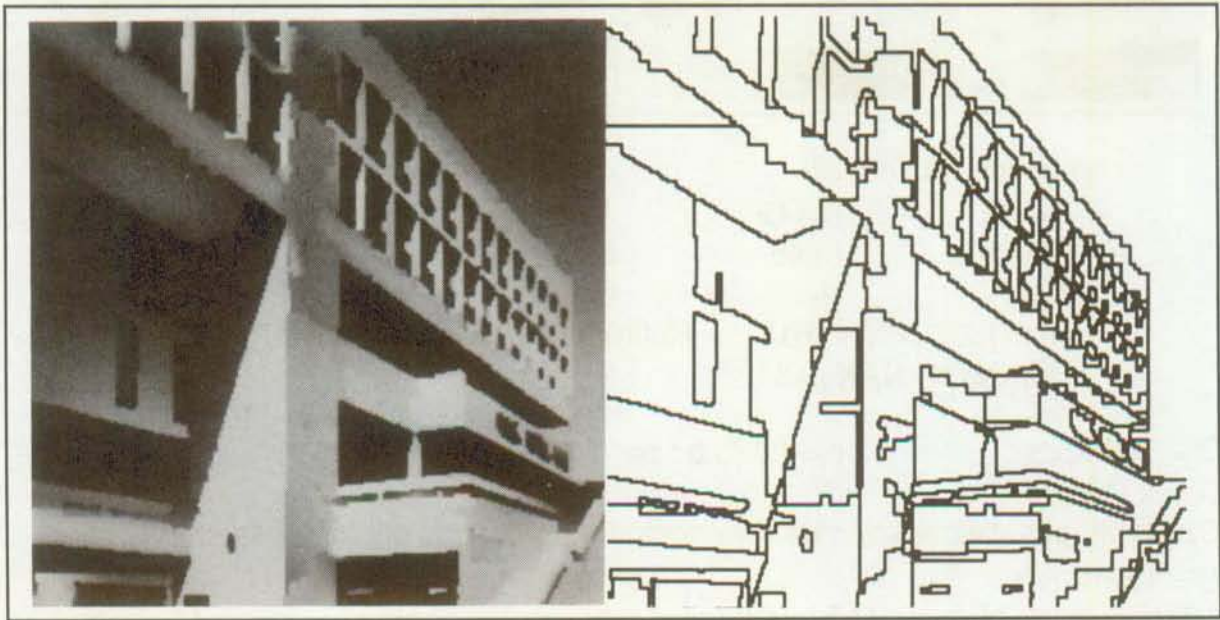
La compression d'information atteint des niveaux très élevés pour l'image de la figure 6.23a (environ 8500 bits suffisent). Comme l'image de la figure 6.14a, on a perdu en qualité par rapport à l'image segmentée en 50 régions. Les déformations apparaissant au niveau de certains contours réels deviennent particulièrement désagréables (partie gauche du menton de la femme). Les mêmes remarques effectuées pour la figure 6.14 s'appliquent en ce qui concerne la perte des détails pertinents pour la reproduction des visages humains. Les yeux sont cependant mieux rendus sur l'image de la figure 6.23a car ils apparaissent plus contrastés.

De tous les résultats du paragraphe 6.2.2, il ressort que certaines frontières des régions obtenues ne suivent plus parfaitement les contours réels des objets préservés par la segmentation. L'augmentation du degré du polynôme de 0 à 1 permet de garder un nombre plus important de détails car il est possible de suivre des surfaces à variation plus complexes. Ceci va être confirmé par les deux paragraphes suivants présentant les résultats de la méthode pour des polynômes de degrés 2 et 3.

6.2.3 Segmentation adaptative par polynômes de degré 2:

Les figures 6.24 à 6.33 correspondent aux résultats de la méthode de codage lorsqu'on utilise une approximation par polynômes de degré $\alpha=2$ pour effectuer la segmentation. Sauf indication contraire, on peut reprendre les remarques faites au paragraphe 6.2.2 pour les images ayant été obtenues avec le même nombre de régions.

Les figures 6.24 à 6.26 montrent les images du bâtiment reconstruites pour des segmentations formées de $N_r=99$, 49 et 24 régions, respectivement. Les facteurs de compression associés sont de 32 à 1, 51 à 1, et 77 à 1. On n'a pas rajouté de signal aléatoire à ces images. Remarquer que dans ces images, la zone homogène correspondant au ciel ne correspond plus qu'à une seule région.



a) Image reconstruite

$\alpha=2$ $N_r=99$
 $M=126$ $L=99$
 $N_0=2$ $N_1=3$
 $\text{Huff}(1,2,\dots,13) = 10853$
 $\text{Huff}(N_0,N_1,N_2,N_3)=104$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=13$ $n_c=7426$
 $N_2=94$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 7942$ $\text{Huff}_0 = 2555$

$C_c = 10679$ bits

$b_c = 1.438$ bit par point contour.

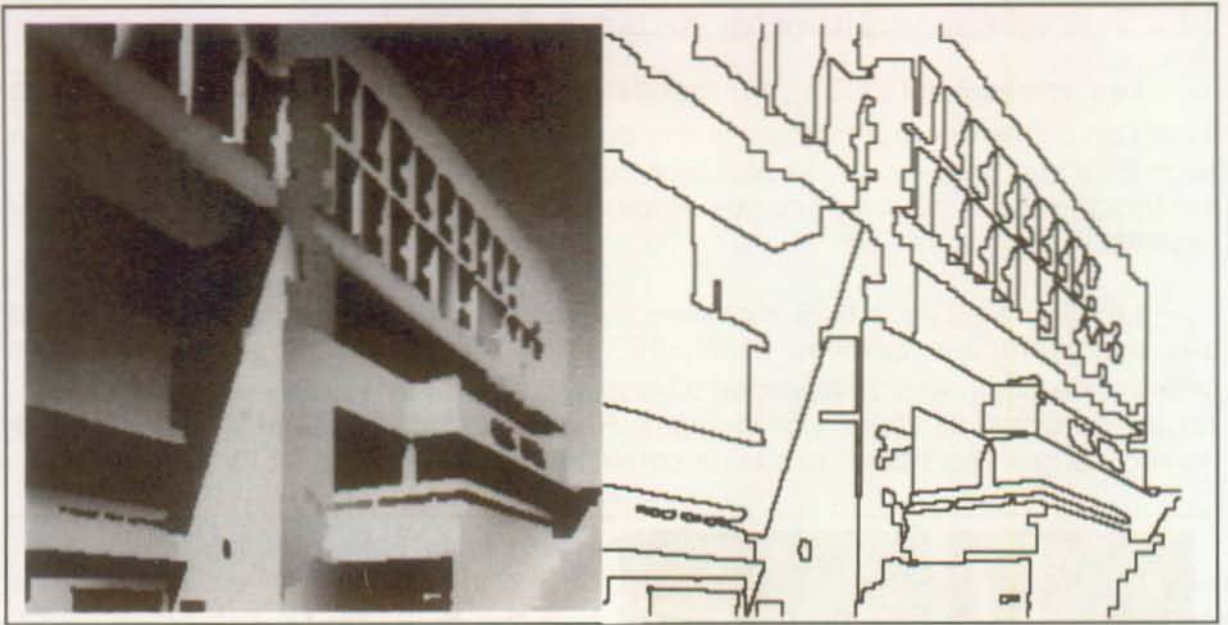
$C_t = 5586$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); pas de composante aléatoire.

$C = 16265$ bits

$\rho = 32.23$

Figure 6.24: Image du bâtiment reconstruite après segmentation en 99 régions.



a) Image reconstruite

$\alpha=2$ $N_r=49$
 $M=63$ $L=49$
 $N_0=1$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 7618$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 51$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=10$ $n_c=5834$
 $N_2=47$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 5345$ $\text{Huff}_0 = 2010$

$C_c = 7495$ bits

$b_c = 1.285$ bit par point contour.

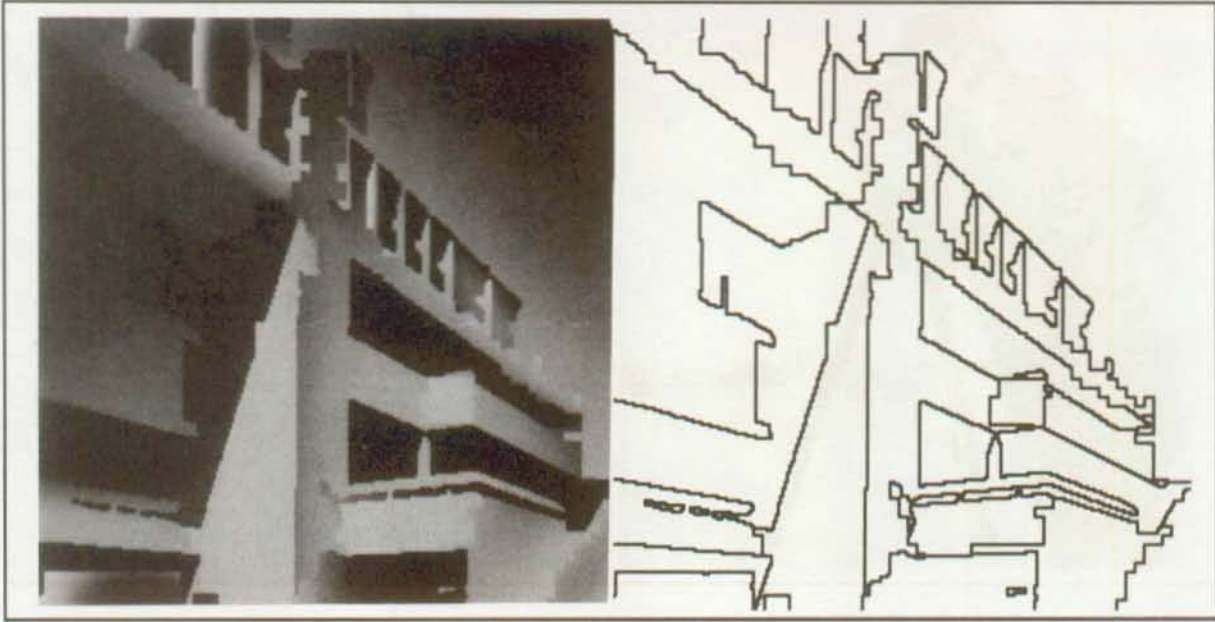
$C_l = 2717$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); pas de composante aléatoire.

$C = 10212$ bits

$\rho = 51.34$

Figure 6.25: Image du bâtiment reconstruite après segmentation en 49 régions.



a) Image reconstruite

$\alpha=2$ $N_r=24$
 $M=32$ $L=24$
 $N_0=0$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 5605$
 $\text{Huff}(N_0,N_1,N_2,N_3)=0$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=6$ $n_c=4222$
 $N_2=24$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 3945$ $\text{Huff}_0 = 1501$

$C_c = 5530$ bits

$b_c = 1.310$ bit par point contour.

$C_t = 1312$ bits

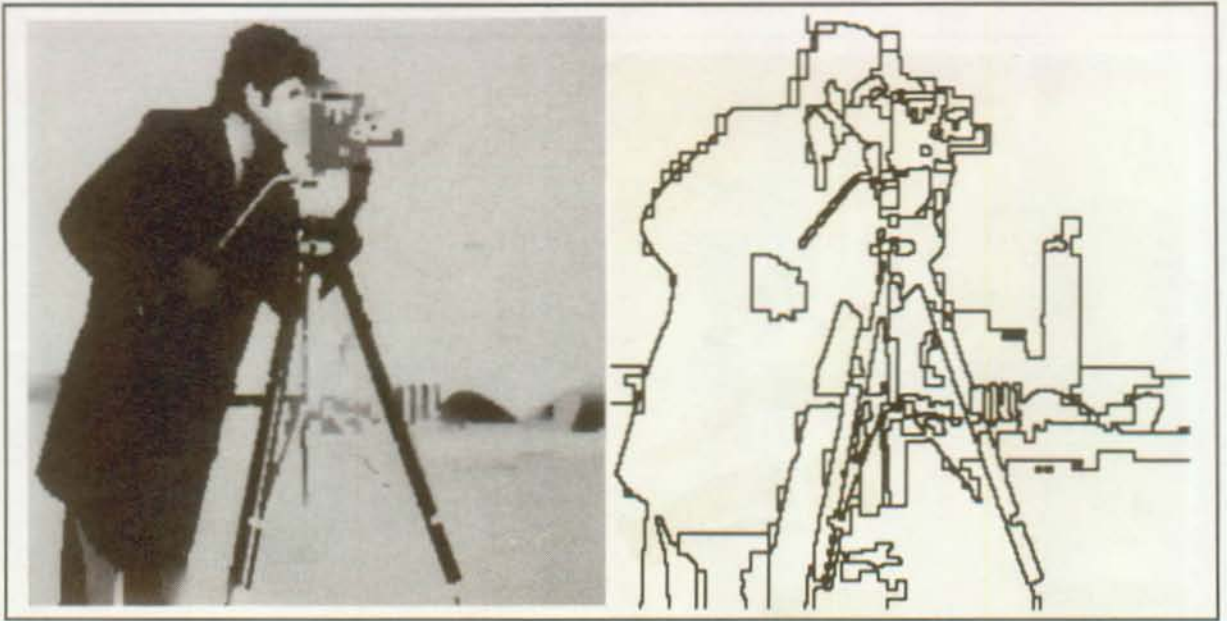
$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); pas de composante aléatoire.

$C = 6842$ bits

$\rho = 76.62$

Figure 6.26: Image du bâtiment reconstruite après segmentation en 24 régions.

Les figures 6.27 à 6.29 montrent les images du caméraman reconstruites pour des segmentations formées de $N_r=99$, 49 et 24 régions, respectivement. Les facteurs de compression associés sont de 37 à 1, 56 à 1 et 85 à 1. On n'a pas rajouté de composante aléatoire aux images reconstruites.



a) Image reconstruite

$\alpha=2$ $N_r=99$
 $M=122$ $L=97$
 $N_0=2$ $N_1=2$
 $\text{Huff}(1,2,\dots,13) = 8273$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 101$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=7$ $n_c=5313$
 $N_2=93$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6607$ $\text{Huff}_0 = 1663$

$C_c = 8588$ bits

$b_c = 1.616$ bit par point contour.

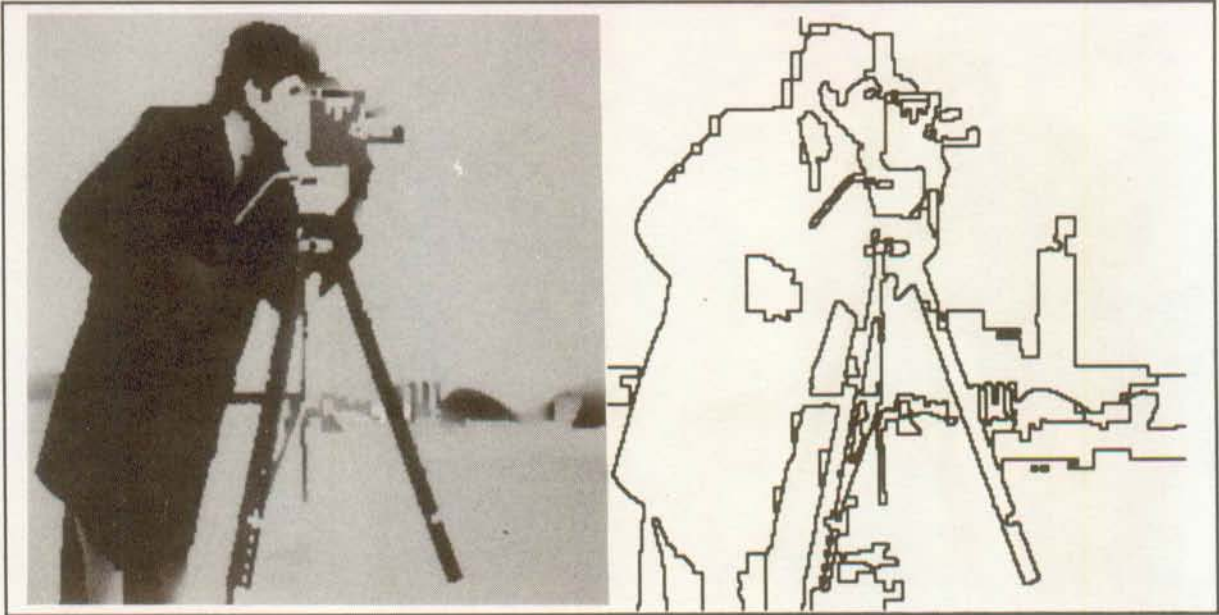
$C_l = 5483$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); pas de composante aléatoire.

$C = 14071$ bits

$\rho = 37.26$

Figure 6.27: Image du caméraman reconstruite après segmentation en 99 régions.



a) Image reconstruite

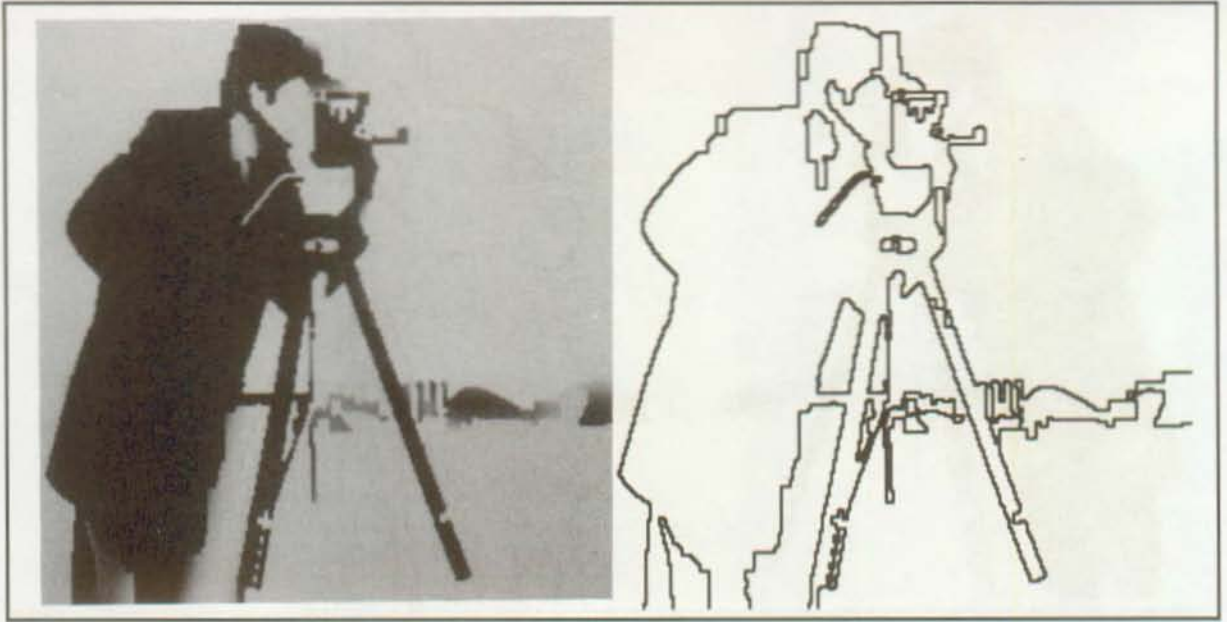
$\alpha=2$ $N_r=49$
 $M=65$ $L=48$
 $N_0=0$ $N_1=2$
 $\text{Huff}(1,2,\dots,13) = 6590$
 $\text{Huff}(N_0,N_1,N_2,N_3)=48$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=10$ $n_c=4341$
 $N_2=46$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 5013$ $\text{Huff}_0 = 1514$

$C_c = 6667$ bits $b_c=1.536$ bit par point contour.
 $C_t = 2694$ bits
 $C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); pas de composante aléatoire.
 $C = 9361$ bits $\rho = 56.01$

Figure 6.28: Image du caméraman reconstruite après segmentation en 49 régions.



a) Image reconstruite

$\alpha=2$ $N_r=24$
 $M=35$ $L=23$
 $N_0=0$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 4777$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 23$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=8$ $n_c=3194$
 $N_2=22$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 3707$ $\text{Huff}_0 = 1083$

$C_c = 4889$ bits

$b_c = 1.531$ bit par point contour.

$C_t = 1278$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); pas de composante aléatoire.

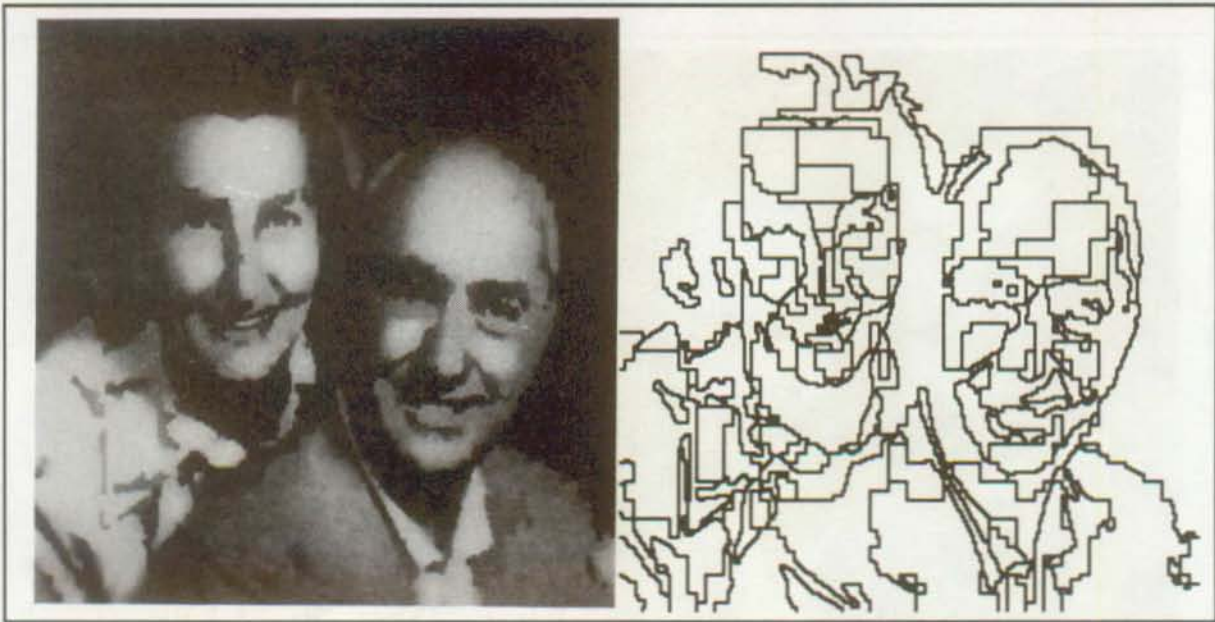
$C = 6167$ bits

$\rho = 85.02$

Figure 6.29: Image du caméraman reconstruite après segmentation en 24 régions.

En ce qui concerne les figures 6.28 et 6.29, il faut remarquer les défauts apparaissant dans la reproduction des contours réels du manteau du caméraman. Ceux-ci ne deviennent vraiment visibles que sur l'image de la figure 6.29a car de petites régions situées de part et d'autre du contour de la figure 6.28b ont été associées à une zone homogène de plus grande taille avoisinante.

Les figures 6.30 à 6.32 montrent les images du couple reconstruites pour des segmentations formées de $N_r=99$, 49 et 24, respectivement. Les facteurs de compression associés sont de 29 à 1, 43 à 1 et 62 à 1. Comme les espaces réguliers correspondent à des surfaces polynomiales du second degré, on a rehaussé la qualité de l'image reconstruite par lissage des faux contours. Pour donner un aspect plus naturel, une composante aléatoire de signal (bruit blanc gaussien d'écart type $\sigma=10$ filtré passe-bas par un moyennneur 3x3) est rajoutée.



a) Image reconstruite

$\alpha=2$ $N_r=99$
 $M=133$ $L=99$
 $N_0=2$ $N_1=3$
 $\text{Huff}(1,2,\dots,13) = 11683$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 104$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=11$ $n_c=7558$
 $N_2=94$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 9069$ $\text{Huff}_0 = 2582$

$C_c = 11805$ bits

$b_c = 1.562$ bit par point contour.

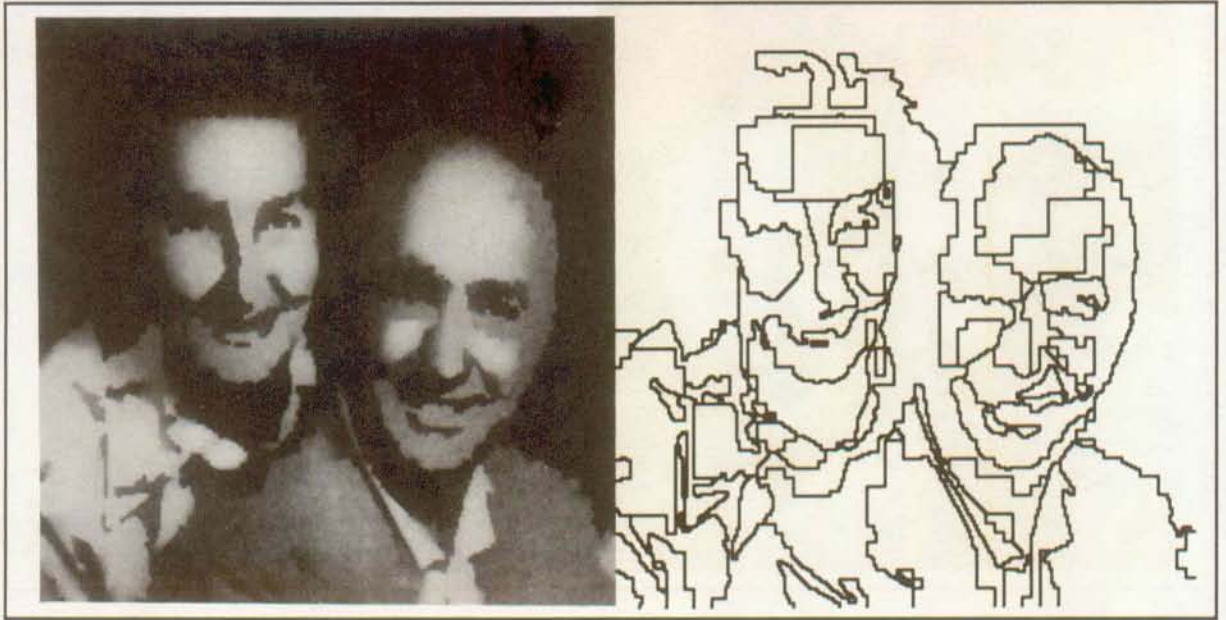
$C_t = 5635$ bits

$C_r = 912$ bits (élimination des faux contours dépassant $d=20$: $n_e=76$); adjonction d'une composante aléatoire.

$C = 18352$ bits

$\rho = 28.56$

Figure 6.30: Image du couple reconstruite après segmentation en 99 régions.



a) Image reconstruite

$\alpha=2$ $N_r=49$
 $M=66$ $L=49$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 8719$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 49$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=9$ $n_c=6037$
 $N_2=48$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6662$ $\text{Huff}_0 = 2103$

$C_c = 8845$ bits

$b_c = 1.465$ bit par point contour.

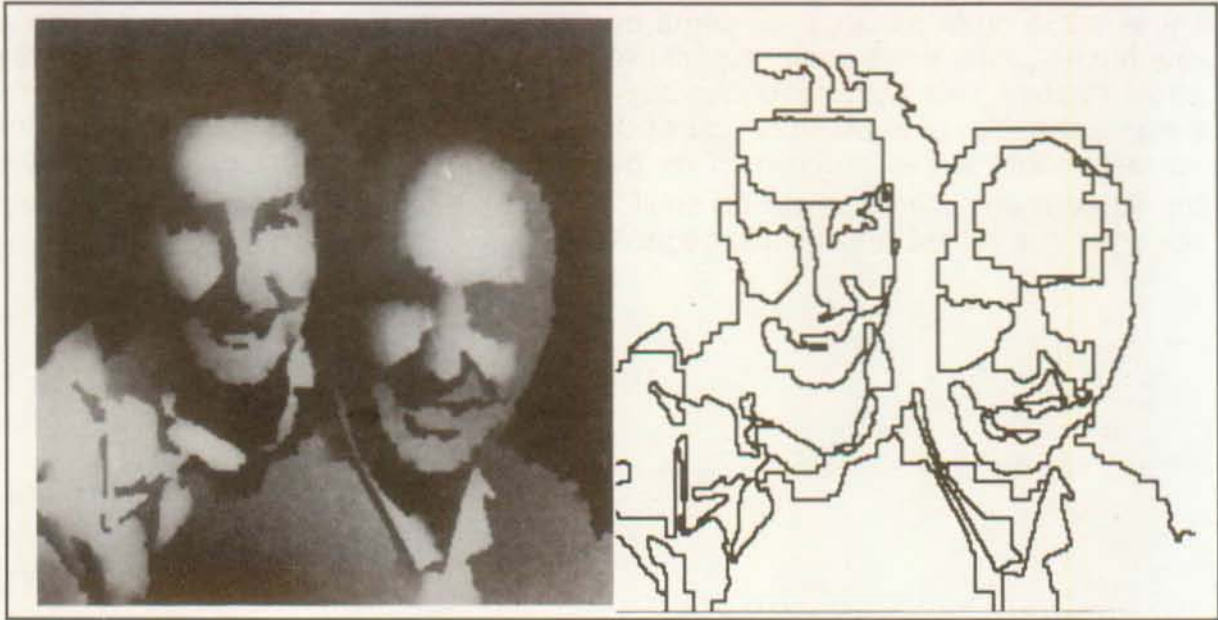
$C_t = 2757$ bits

$C_r = 588$ bits (élimination des faux contours dépassant $d=20:n_e=49$); adjonction d'une composante aléatoire.

$C = 12190$ bits

$\rho = 43.01$

Figure 6.31: Image du couple reconstruite après segmentation en 49 régions.



a) Image reconstruite

$\alpha=2$ $N_r=24$
 $M=34$ $L=24$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 6665$
 $\text{Huff}(N_0,N_1,N_2,N_3)=24$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=8$ $n_c=4828$
 $N_2=23$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 5086$ $\text{Huff}_0 = 1744$

$C_c = 6777$ bits

$b_c = 1.404$ bit par point contour.

$C_t = 1306$ bits

$C_r = 348$ bits (élimination des faux contours dépassant $d=20:n_e=29$); adjonction d'une composante aléatoire.

$C = 8431$ bits

$\rho = 62.19$

Figure 6.32: Image du couple reconstruite après segmentation en 24 régions.

Si l'on compare les figures 6.31 et 6.32 aux résultats calculés pour l'image du couple avec des polynômes d'ordre 0 et 1, on peut observer que les régions correspondant à des surfaces de variation parabolique telles les fronts des personnages sont mieux représentés dans le premier cas, ce qui confirme la bonne adaptation des formes des régions aux paramètres de l'approximation en fonction des données à traiter. Ceci permet de préserver un certain nombre de détails mais des distorsions apparaissent dans la reproduction de certains contours. On observe ainsi des défauts non négligeables entre l'épaule gauche de l'homme et l'arrière plan de l'image.

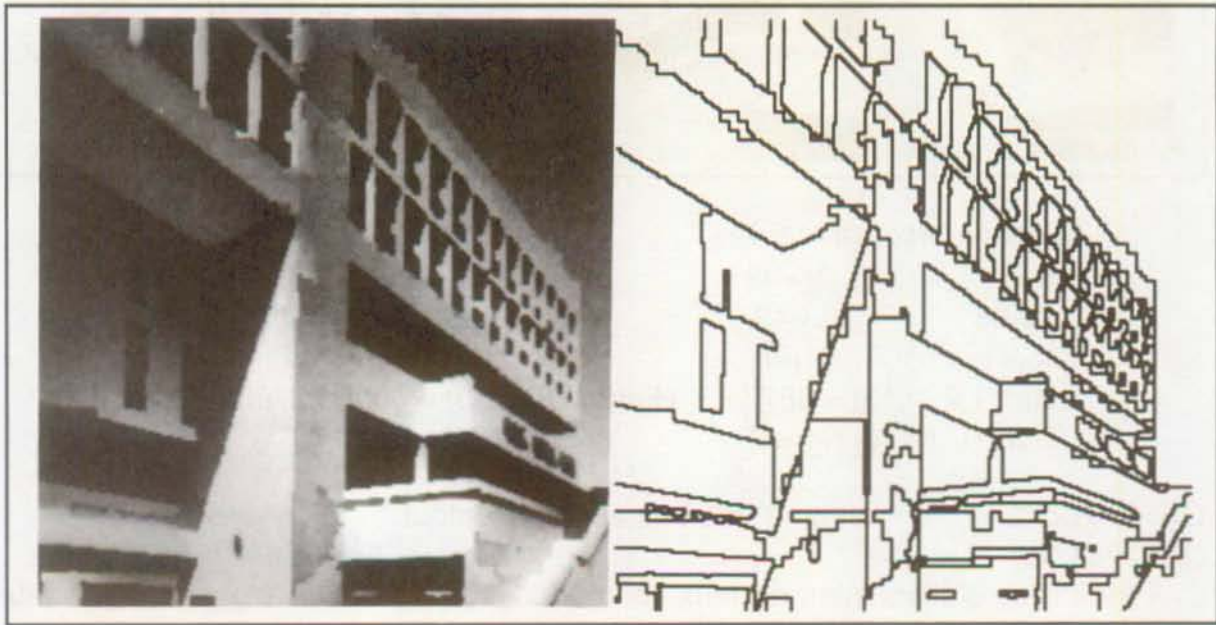
De tous les résultats du paragraphe 6.2.3, il ressort que l'augmentation du degré du polynôme confirme la tendance observée au paragraphe 6.2.2. Ainsi, il est possible à compression (presque) égale de suivre des surfaces à variation plus complexes au détriment de la précision dans la reproduction de certains contours réels de l'image. La diminution du nombre de régions nécessaires à la reproduction de surfaces à variation complexes évite de faire disparaître lors du processus de rassemblement d'autres régions pouvant avoir une signification importante (par exemple, bouche des personnages dans l'image du couple). L'effet

sera d'autant plus visible dans les images qui vont suivre notamment l'image de la figure 6.35a où le passage du degré du polynôme de 2 à 3 permet de préserver une bonne partie de la partie supérieure droite de l'image du bâtiment segmentée en 24 régions. Une autre remarque que l'on peut faire à la vue des résultats de ce paragraphe est que presque toutes les surfaces de reconstruction obtenues correspondent à des polynômes de degré 2 indépendamment du type d'image traitée pour une même valeur du seuil T (5^2). En effet, dans tous les cas on peut observer que N_2 est sensiblement égal à L .

6.2.4 Segmentation adaptative par polynômes de degré 3:

Les figures 6.33 à 6.43 correspondent aux résultats de la méthode de codage lorsqu'on utilise une approximation par surfaces polynomiales de degré $\alpha=3$.

Les figures 6.33 à 6.35 montrent les images du bâtiment reconstruites pour des segmentations formées de $N_r=99$, 49 et 24 régions, respectivement. Les facteurs de compression associés sont de 32 à 1, 45 à 1, et 66 à 1. On a rajouté de façon globale aux deux images reconstruites des figures 6.33a et 6.34a respectivement la même composante aléatoire que pour les reconstructions des images du couple.



a) Image reconstruite

$\alpha=3$ $N_r=99$
 $M=128$ $L=99$
 $N_0=1$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 11654$
 $\text{Huff}(N_0,N_1,N_2,N_3)=101$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=13$ $n_c=7483$
 $N_2=97$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 8440$ $\text{Huff}_0 = 2749$

$C_c = 10886$ bits

$b_c = 1.455$ bit par point contour.

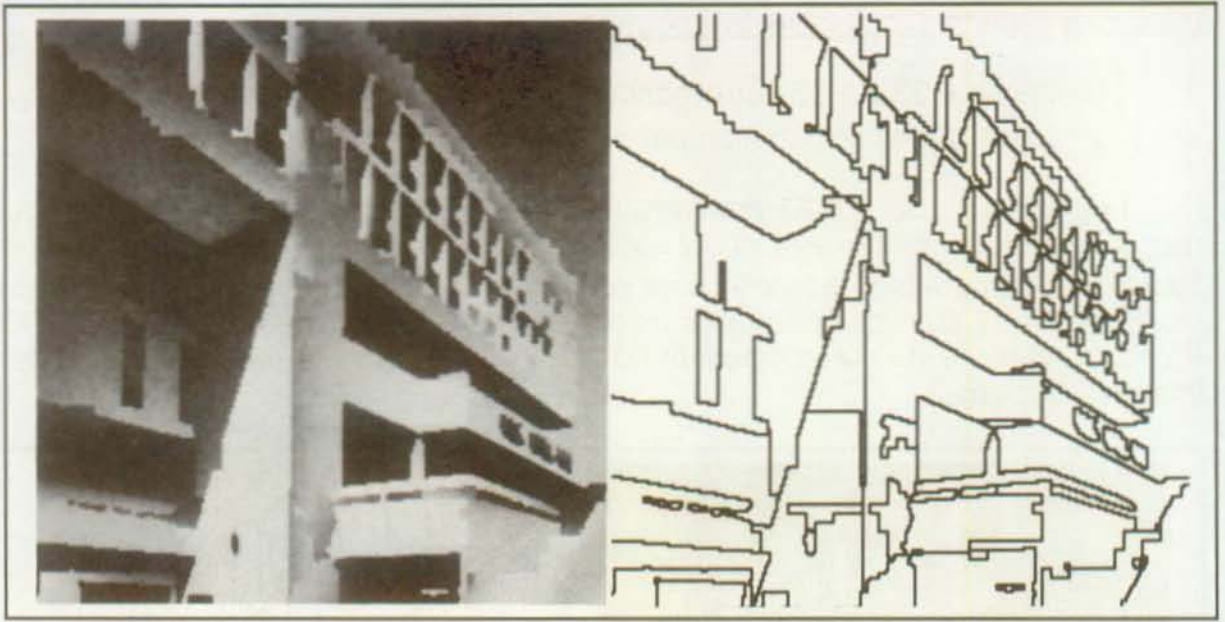
$C_t = 5685$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); adjonction d'une composante aléatoire.

$C = 16571$ bits

$\rho = 31.64$

Figure 6.33: Image du bâtiment reconstruite après segmentation en 99 régions.



a) Image reconstruite

$\alpha=3$ $N_r=49$
 $M=65$ $L=49$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 8877$
 $\text{Huff}(N_0,N_1,N_2,N_3)=49$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=12$ $n_c=6354$
 $N_2=48$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6458$ $\text{Huff}_0 = 2155$

$C_c = 8781$ bits

$b_c = 1.382$ bit par point contour.

$C_t = 2751$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); adjonction d'une composante aléatoire.

$C = 11532$ bits

$\rho = 45.46$

Figure 6.34: Image du bâtiment reconstruite après segmentation en 49 régions.



a) Image reconstruite

$\alpha=3$ $N_r=24$
 $M=41$ $L=24$
 $N_0=0$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 6724$
 $\text{Huff}(N_0,N_1,N_2,N_3)=24$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=7$ $n_c=5117$
 $N_2=24$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4752$ $\text{Huff}_0 = 1753$

$C_c = 6603$ bits

$b_c = 1.290$ bit par point contour.

$C_l = 1357$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); pas de composante aléatoire.

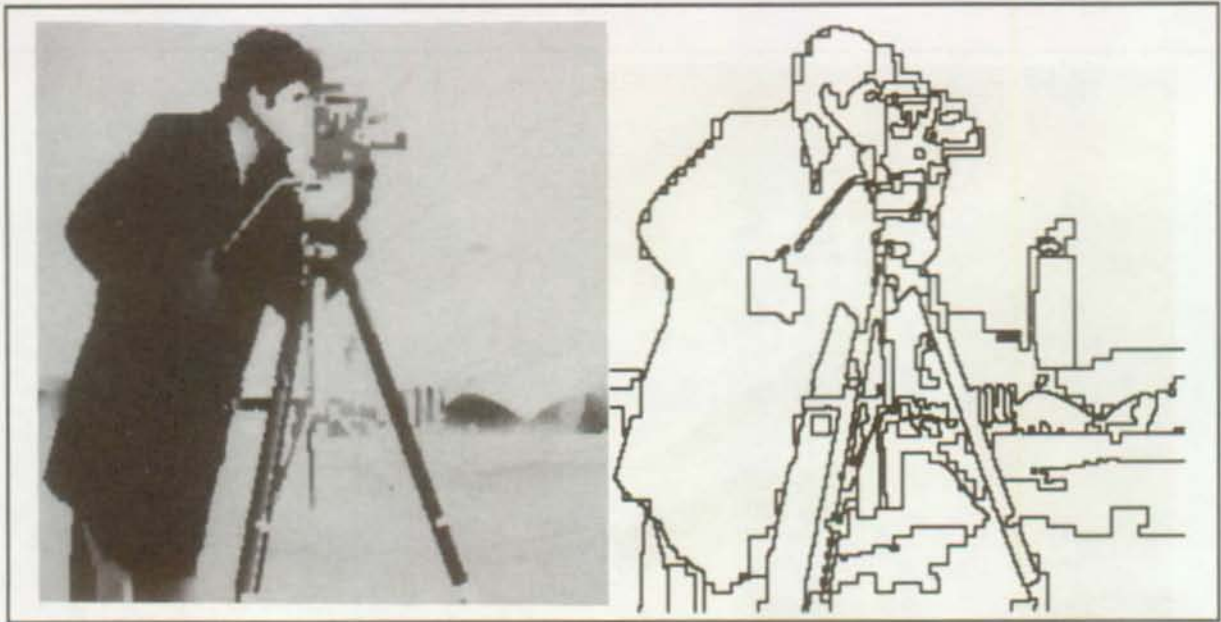
$C = 7960$ bits

$\rho = 65.87$

Figure 6.35: Image du bâtiment reconstruite après segmentation en 24 régions.

Si la qualité de l'image du bâtiment segmentée en 24 régions avec des polynômes de degré 3 (figure 6.35a) est meilleure que la qualité des images obtenues avec des polynômes de degré inférieur mais pour un même nombre de régions (figures 6.7a, 6.17a et 6.26a), cela se paie par un nombre plus élevé de points contours à coder (5117 au lieu de 3901, 4605 et 4222, respectivement) d'où une légère baisse du facteur de compression.

Les figures 6.36 à 6.39 montrent les images du caméraman reconstruites pour des segmentations formées de $N_r=99$, 49, 24 et 9 régions, respectivement. Les facteurs de compression associés sont de 36 à 1, 54 à 1, 79 à 1 et 132 à 1. Une composante aléatoire a été rajoutée au signal approximé par des polynômes.



a) Image reconstruite

$\alpha=3$ $N_r=99$
 $M=125$ $L=98$
 $N_0=3$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 9231$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 102$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=8$ $n_c=5705$
 $N_2=94$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 7159$ $\text{Huff}_0 = 1947$

$C_c = 9210$ bits

$b_c = 1.614$ bit par point contour.

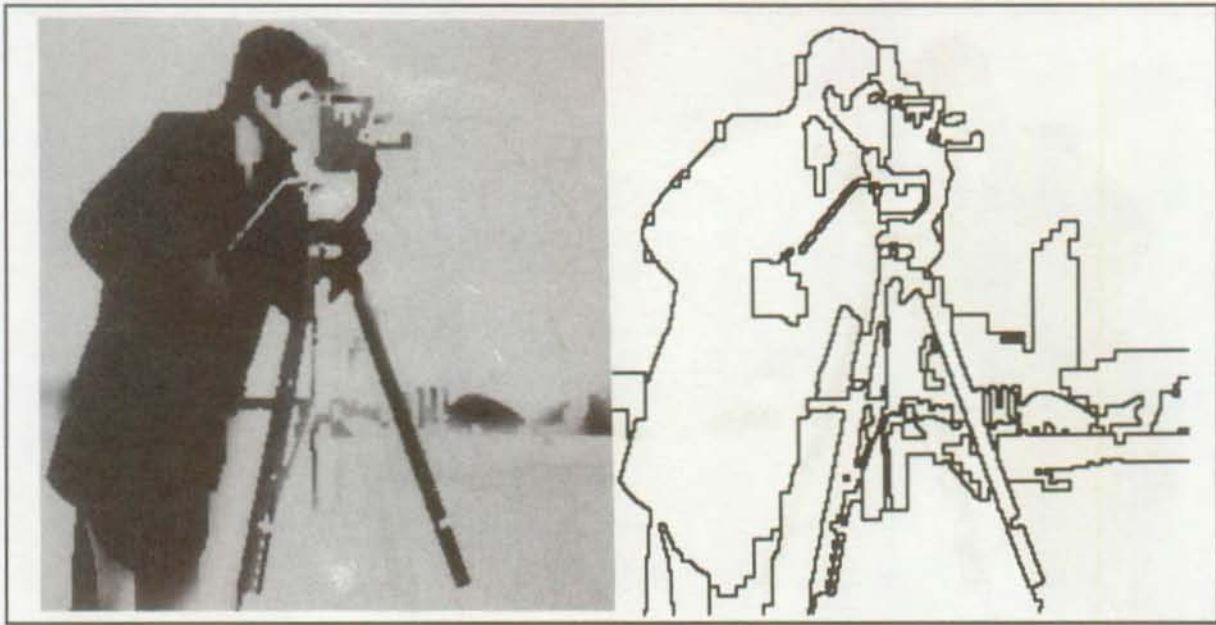
$C_t = 5537$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); adjonction d'une composante aléatoire.

$C = 14747$ bits

$\rho = 35.55$

Figure 6.36: Image du caméraman reconstruite après segmentation en 99 régions.



a) Image reconstruite

$\alpha=3$ $N_r=49$
 $M=68$ $L=49$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 6900$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 49$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=7$ $n_c=4474$
 $N_2=48$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 5291$ $\text{Huff}_0 = 1515$

$C_c = 6904$ bits

$b_c = 1.543$ bit par point contour.

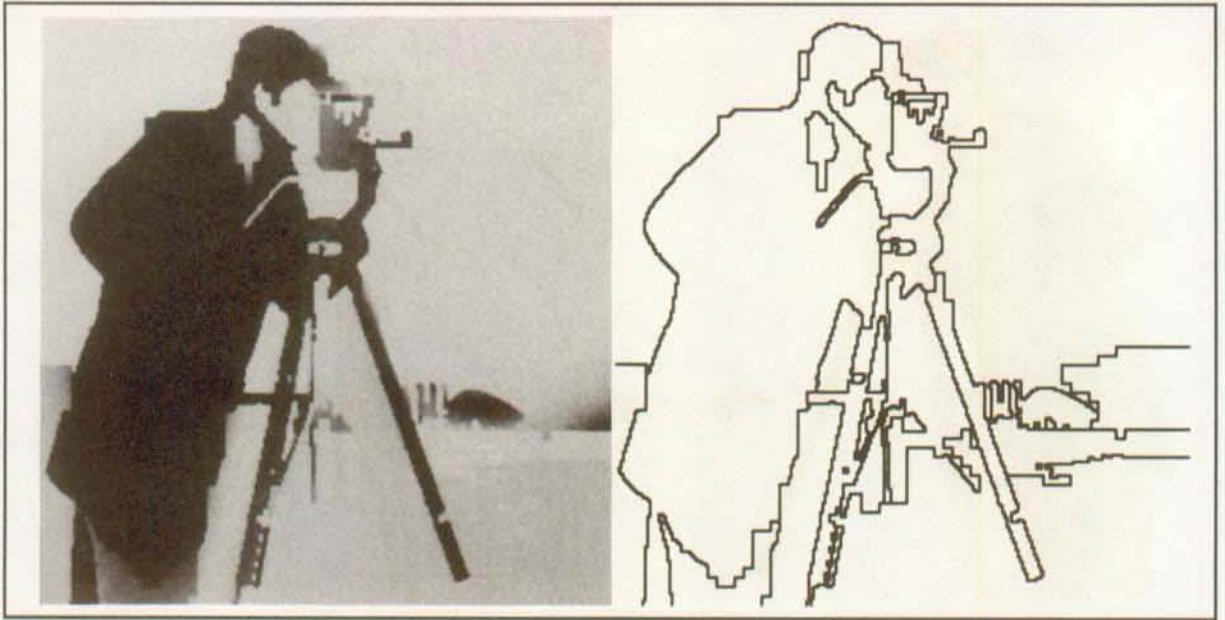
$C_t = 2769$ bits

$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$); adjonction d'une composante aléatoire.

$C = 9673$ bits

$\rho = 54.20$

Figure 6.37: Image du caméraman reconstruite après segmentation en 49 régions.

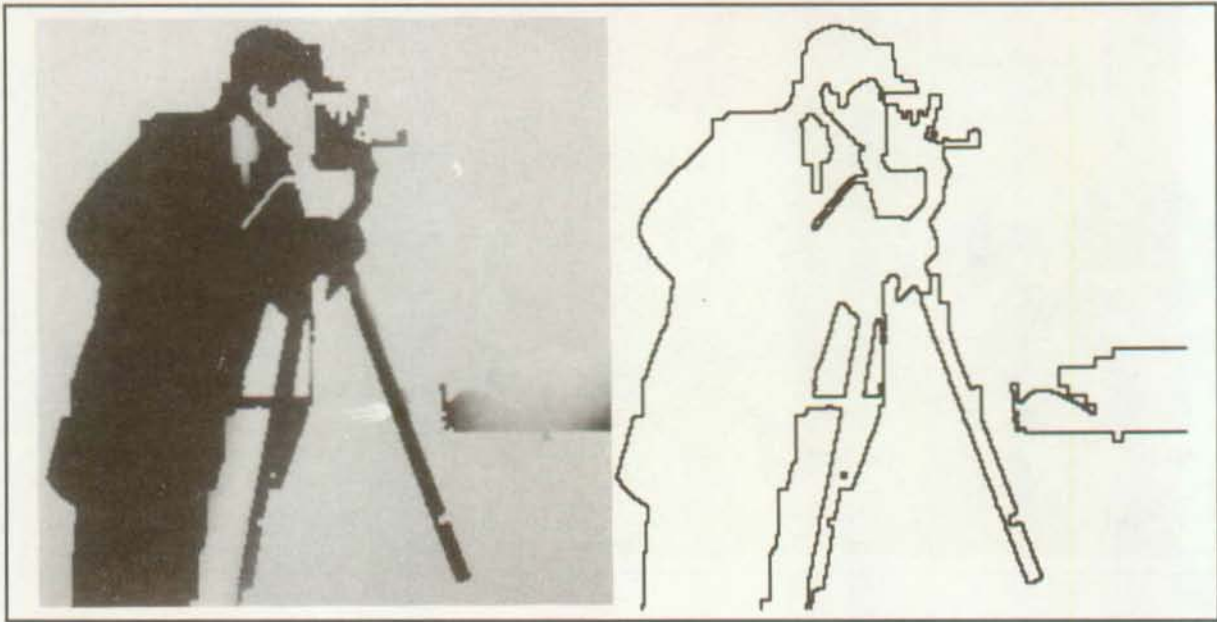


a) Image reconstruite

 $\alpha=3$ $N_r=24$ $M=42$ $L=24$ $N_0=1$ $N_1=1$ $\text{Huff}(1,2,\dots,13) = 5238$ $\text{Huff}(N_0, N_1, N_2, N_3) = 29$

b) Image interfrontière 4-connexe.

 $A=499$ $T=5^2$ $p=6$ $n_c=3525$ $N_2=21$ $N_3=1$ $\text{Huff}(0,2,3,\dots,13) = 3992$ $\text{Huff}_0 = 1202$ $C_c = 5278$ bits $b_c = 1.497$ bit par point contour. $C_t = 1359$ bits $C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire. $C = 6637$ bits $\rho = 79.00$ Figure 6.38: Image du caméraman reconstruite après segmentation en 24 régions.



a) Image reconstruite

$\alpha=3$ $N_r=9$
 $M=42$ $L=9$
 $N_0=0$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 3387$
 $\text{Huff}(N_0,N_1,N_2,N_3)=9$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=6$ $n_c=2438$
 $N_2=8$ $N_3=1$
 $\text{Huff}(0,2,3,\dots,13) = 2539$ $\text{Huff}_0 = 816$

$C_c = 3439$ bits

$b_c = 1.411$ bit par point contour.

$C_t = 529$ bits

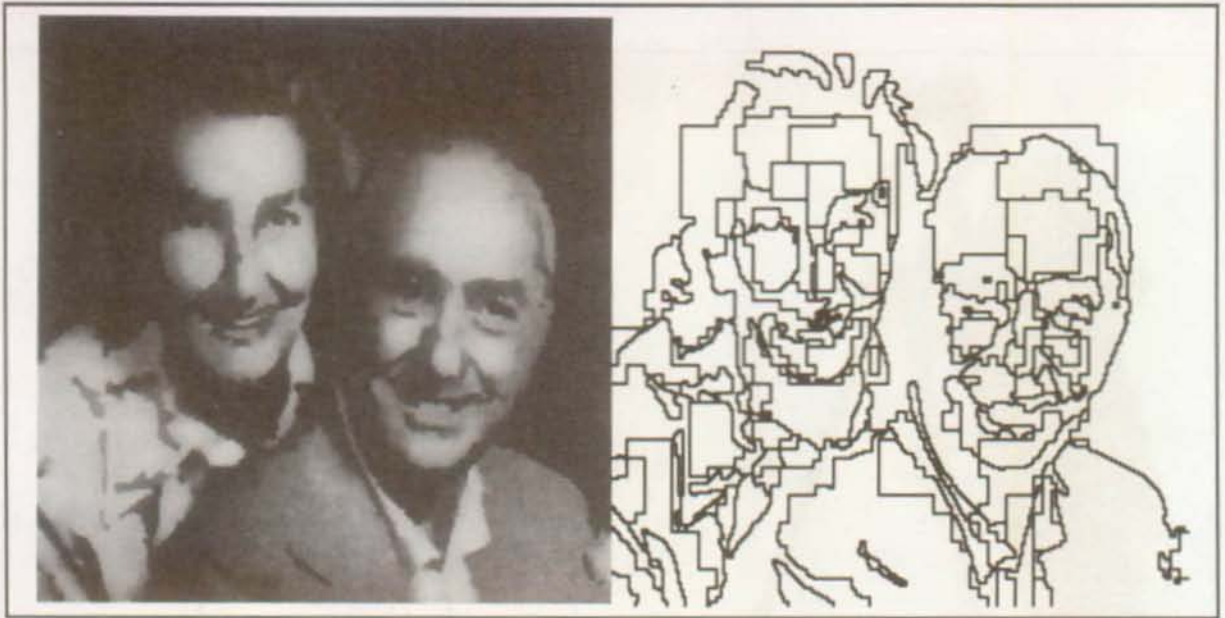
$C_r = 0$ bit (pas d'élimination de faux contours: $n_e=0$) ; pas de composante aléatoire.

$C = 3968$ bits

$\rho = 132.13$

Figure 6.39: Image du caméraman reconstruite après segmentation en 9 régions.

Les figures 6.40 à 6.43 montrent les images du couple reconstruites pour des segmentations formées de $N_r=99, 49, 24$ et 15 respectivement. Les facteurs de compression associés sont de 28 à 1 , 43 à 1 , 63 à 1 et 75 à 1 . Comme les espaces réguliers correspondent à des surfaces polynomiales du second degré, on a rehaussé la qualité de l'image reconstruite par lissage des faux contours. Pour donner un aspect plus naturel, la composante aléatoire décrite précédemment est ajoutée.



a) Image reconstruite

$\alpha=3$ $N_r=99$
 $M=138$ $L=99$
 $N_0=1$ $N_1=2$
 $\text{Huff}(1,2,\dots,13) = 11952$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 102$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=9$ $n_c=7727$
 $N_2=96$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 9275$ $\text{Huff}_0 = 2594$

$C_c = 11995$ bits

$b_c = 1.552$ bit par point contour.

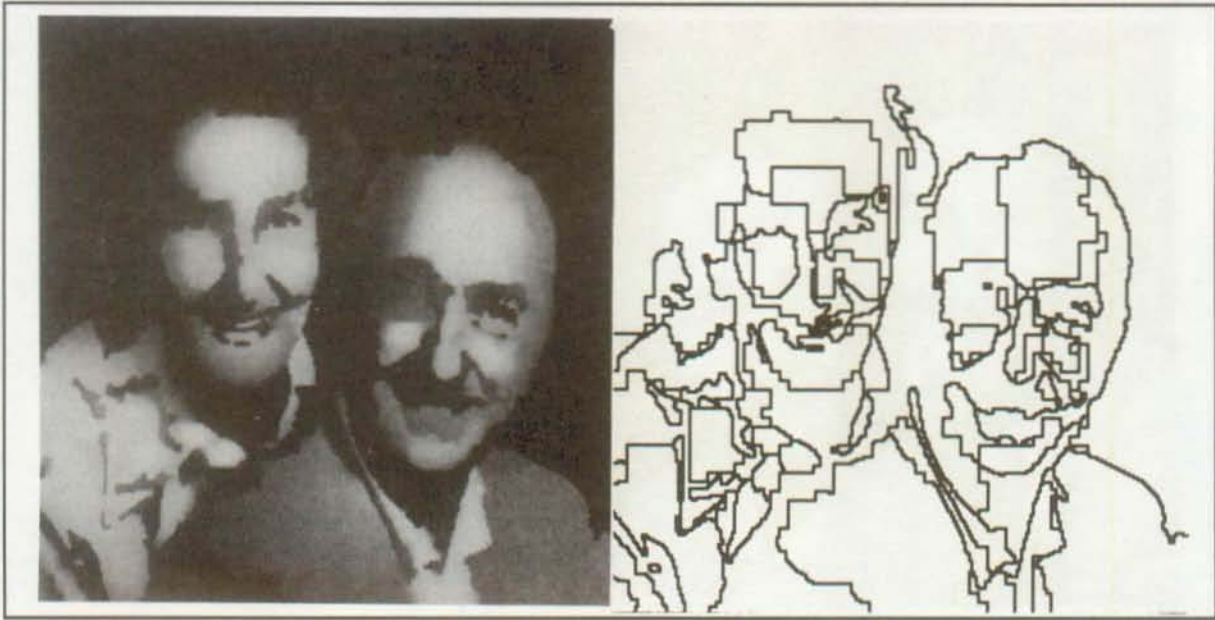
$C_t = 5732$ bits

$C_r = 948$ bits (élimination des faux contours dépassant $d=20$: $n_e=79$); adjonction d'une composante aléatoire.

$C = 18675$ bits

$\rho = 28.07$

Figure 6.40: Image du couple reconstruite après segmentation en 99 régions.



a) Image reconstruite

$\alpha=3$ $N_r=49$
 $M=77$ $L=49$
 $N_0=0$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 8786$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 49$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=9$ $n_c=6015$
 $N_2=48$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6881$ $\text{Huff}_0 = 2021$

$C_c = 8912$ bits

$b_c = 1.482$ bit par point contour.

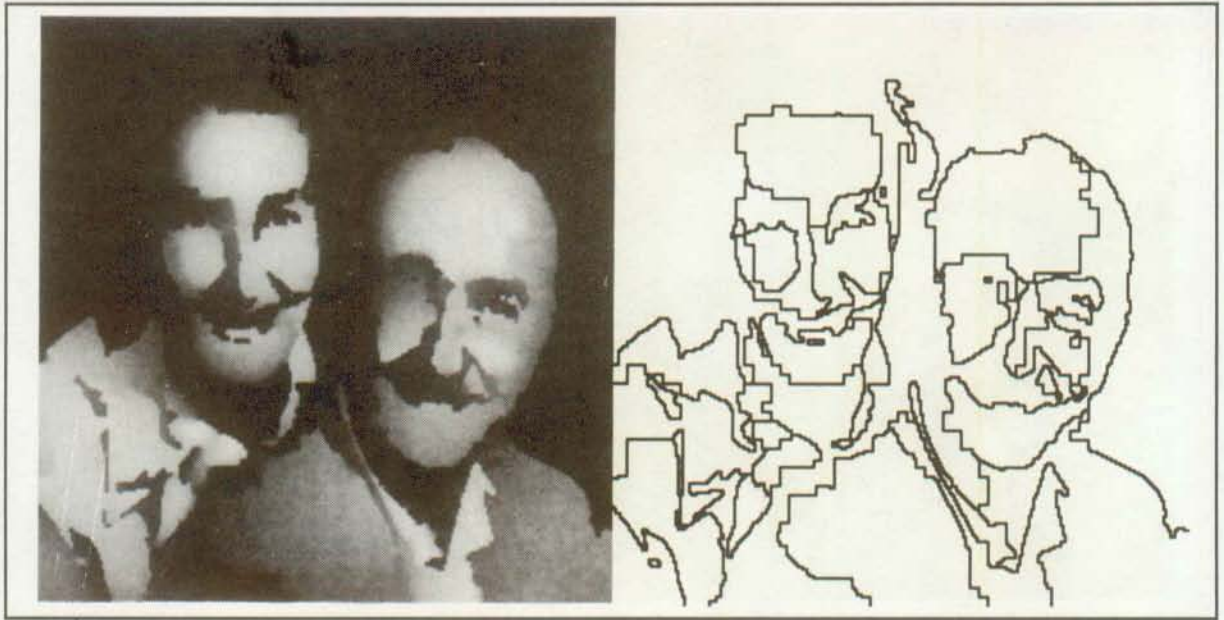
$C_t = 2839$ bits

$C_r = 504$ bits (élimination des faux contours dépassant $d=20$: $n_e=42$); adjonction d'une composante aléatoire.

$C = 12255$ bits

$\rho = 42.78$

Figure 6.41: Image du couple reconstruite après segmentation en 49 régions.



a) Image reconstruite

$$\alpha=3 \quad N_r=24$$

$$M=38 \quad L=24$$

$$N_0=0 \quad N_1=0$$

$$\text{Huff}(1,2,\dots,13) = 6622$$

$$\text{Huff}(N_0, N_1, N_2, N_3) = 0$$

b) Image interfrontière 4-connexe.

$$A=499 \quad T=5^2$$

$$p=7 \quad n_c=4672$$

$$N_2=24 \quad N_3=0$$

$$\text{Huff}(0,2,3,\dots,13) = 5254 \quad \text{Huff}_0 = 1600$$

$$C_c = 6720 \text{ bits}$$

$$b_c = 1.438 \text{ bit par point contour.}$$

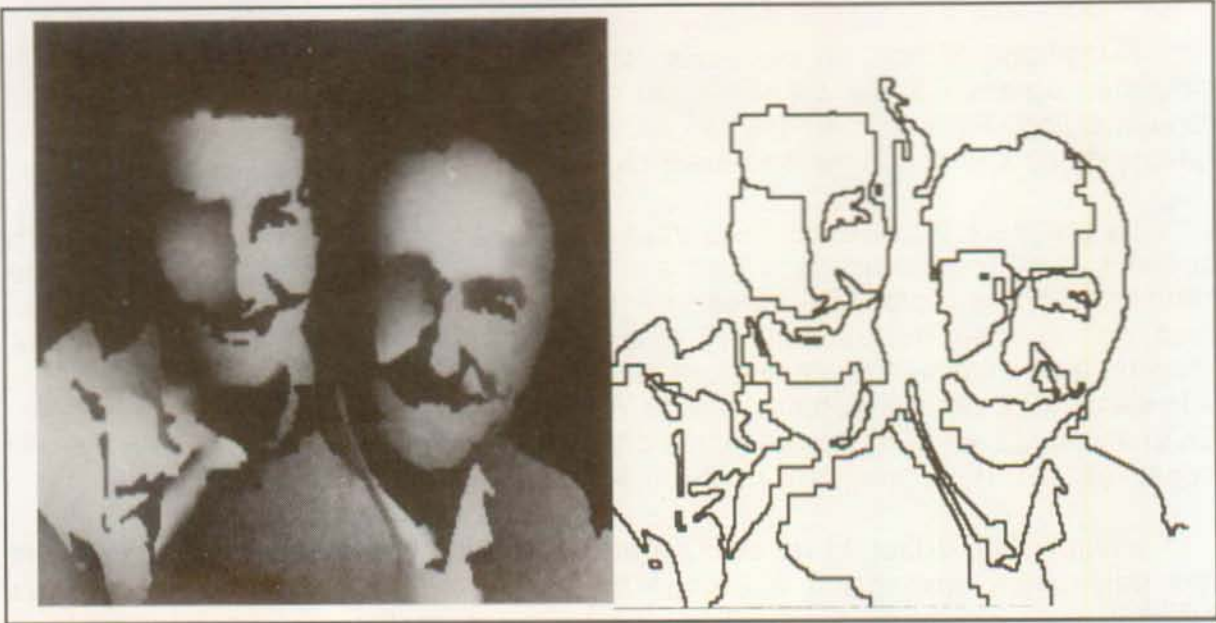
$$C_t = 1342 \text{ bits}$$

$C_r = 204 \text{ bits}$ (élimination des faux contours dépassant $d=20: n_e=17$); adjonction d'une composante aléatoire.

$$C = 8266 \text{ bits}$$

$$\rho = 63.42$$

Figure 6.42: Image du couple reconstruite après segmentation en 24 régions.



a) Image reconstruite

$\alpha=3$ $N_r=15$
 $M=28$ $L=15$
 $N_0=0$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 5875$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 0$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=11$ $n_c=4237$
 $N_2=15$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4669$ $\text{Huff}_0 = 1464$

$C_c = 6029$ bits

$b_c = 1.423$ bit par point contour.

$C_t = 832$ bits

$C_r = 132$ bits (élimination des faux contours dépassant $d=20:n_e=11$); adjonction d'une composante aléatoire.

$C = 6993$ bits

$\rho = 74.97$

Figure 6.43: Image du couple reconstruite après segmentation en 15 régions.

L'image de la figure 6.43 présente une compression élevée pour l'image du couple bien qu'on ait perdu les yeux droits des personnages.

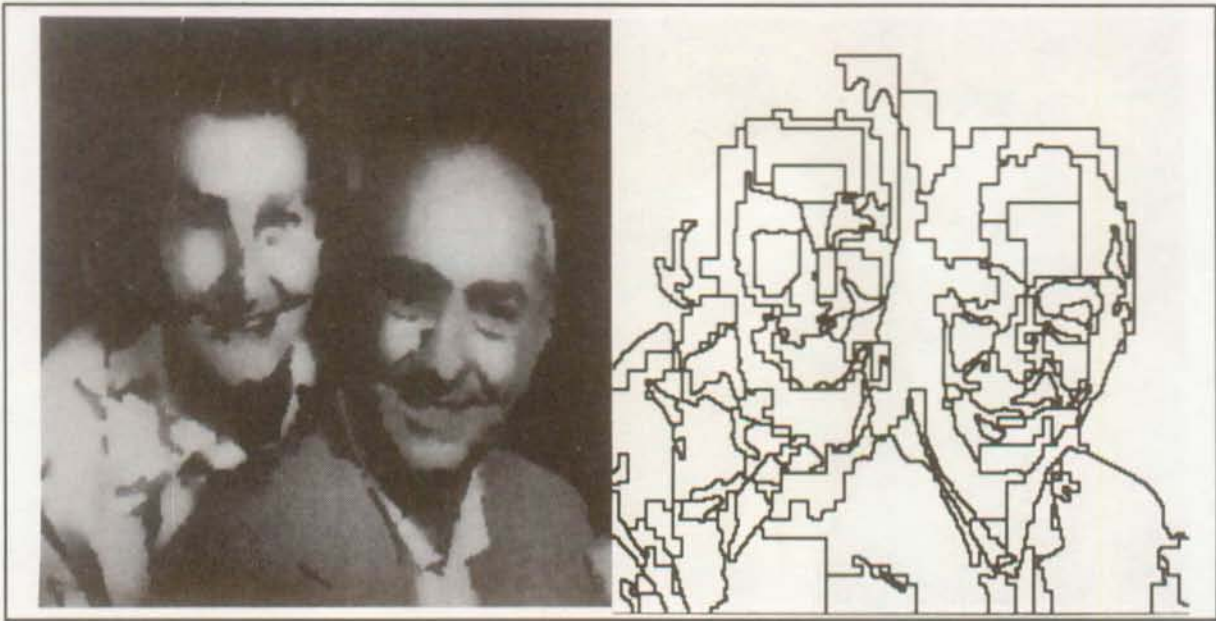
Dans l'ensemble, les résultats de ce paragraphe confirment ceux énoncés au paragraphe 6.2.3.

6.3 Effet du prétraitement de l'image originale :

Dans cette section, on va étudier l'effet de l'utilisation d'une image de contrôle simplifiée obtenue après prétraitement de l'image originale par un opérateur de régularisation. Pour illustrer les résultats de la méthode, on a choisi de prendre l'image du couple et comme opérateur de régularisation le filtre de Nagao.

La première étape consiste à trouver les images de contrôle simplifiées. On applique ainsi l'opérateur de Nagao à l'image originale du couple. Le résultat de cette opération est présenté après une itération à la figure 2.12c. Le filtre passe-haut de réponse fréquentielle (2.1) est ensuite appliqué à cette image. La procédure de détection des passages par zéro et de poursuite des contours définie à la section 2.2 permet d'obtenir l'image de contrôle des contours de la figure 2.22. En analysant le signe de l'image filtrée on obtient l'image des structures positives et négatives. Ces deux images vont être utilisées pour piloter la segmentation.

L'algorithme adaptatif de division et rassemblement est alors appliqué avec des polynômes de degré 3 à l'image du couple originale avec les paramètres prenant leur valeur par défaut. Les images reconstruites correspondantes sont présentées aux figures 6.44 à 6.46 pour des segmentations formées de 99, 49 et 24 régions. On a rajouté la composante aléatoire globale aux surfaces polynomiales. Les facteurs de compression respectifs sont de 30 à 1, 46 à 1 et 64 à 1.



a) Image reconstruite

$\alpha=3$ $N_r=99$
 $M=126$ $L=99$
 $N_0=1$ $N_1=2$
 $\text{Huff}(1,2,\dots,13) = 11952$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 102$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=2$ $n_c=7005$
 $N_2=96$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 8368$ $\text{Huff}_0 = 2384$

$C_c = 10780$ bits

$b_c = 1.539$ bit par point contour.

$C_t = 5648$ bits

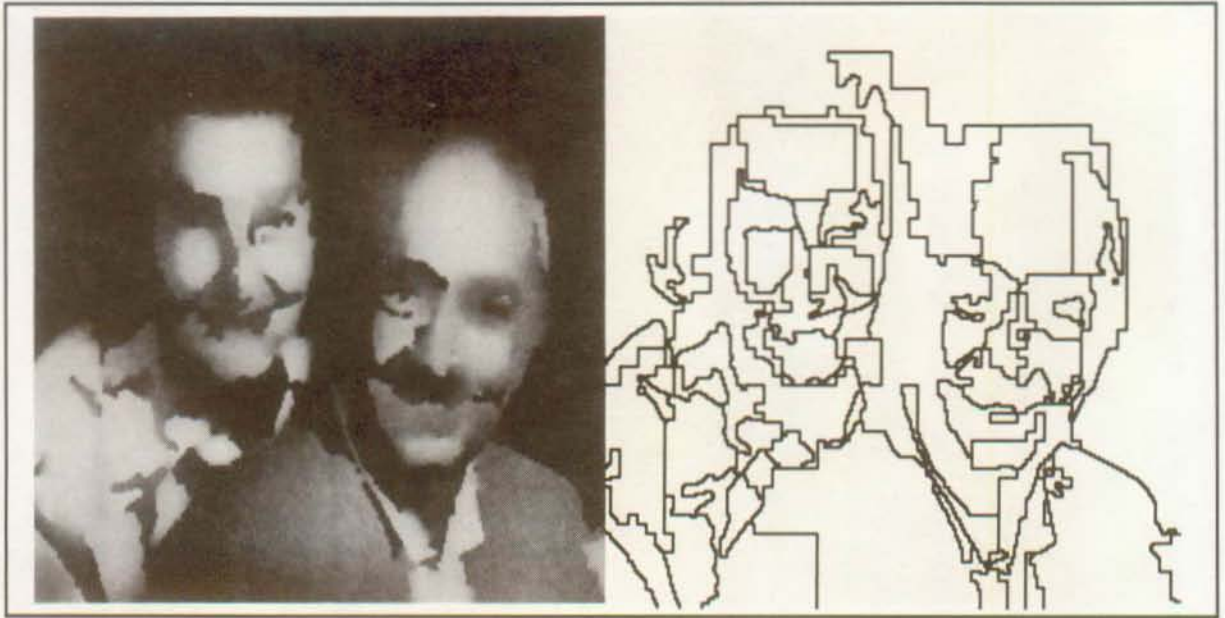
$C_r = 816$ bits (élimination des faux contours dépassant $d=20$: $n_e=68$); adjonction d'une composante aléatoire.

$C = 18675$ bits

$\rho = 30.40$

Figure 6.44: Image du couple reconstruite après segmentation en 99 régions.

Pour mesurer l'effet de l'utilisation des images de contrôle simplifiées, il faut comparer les résultats des figures 6.44 à 6.46 à ceux des figures 6.40 à 6.42, respectivement. Ainsi, si l'on compare les figures 6.40 et 6.44, on remarque que certains détails sont très mal reproduits (yeux de la femme, motifs sur sa chemise). Ces imperfections existant dans les images de contrôle utilisées ne peuvent qu'être reproduites au niveau de la forme des régions de la segmentation. En contrepartie, cette simplification entraîne une diminution de la complexité des contours réels des images et par conséquent des frontières des régions. Le nombre de points contour est ainsi passé de 7727 à 7005. Il en résulte une baisse de l'information contour à coder et par là, le facteur de compression ne peut qu'augmenter. Mais le jeu en vaut-il la chandelle sur le plan de la qualité?



a) Image reconstruite

$\alpha=3$ $N_r=49$
 $M=68$ $L=49$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 8144$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 49$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=5$ $n_c=5615$
 $N_2=48$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6049$ $\text{Huff}_0 = 1929$

$C_c = 8034$ bits

$b_c = 1.433$ bit par point contour.

$C_t = 2720$ bits

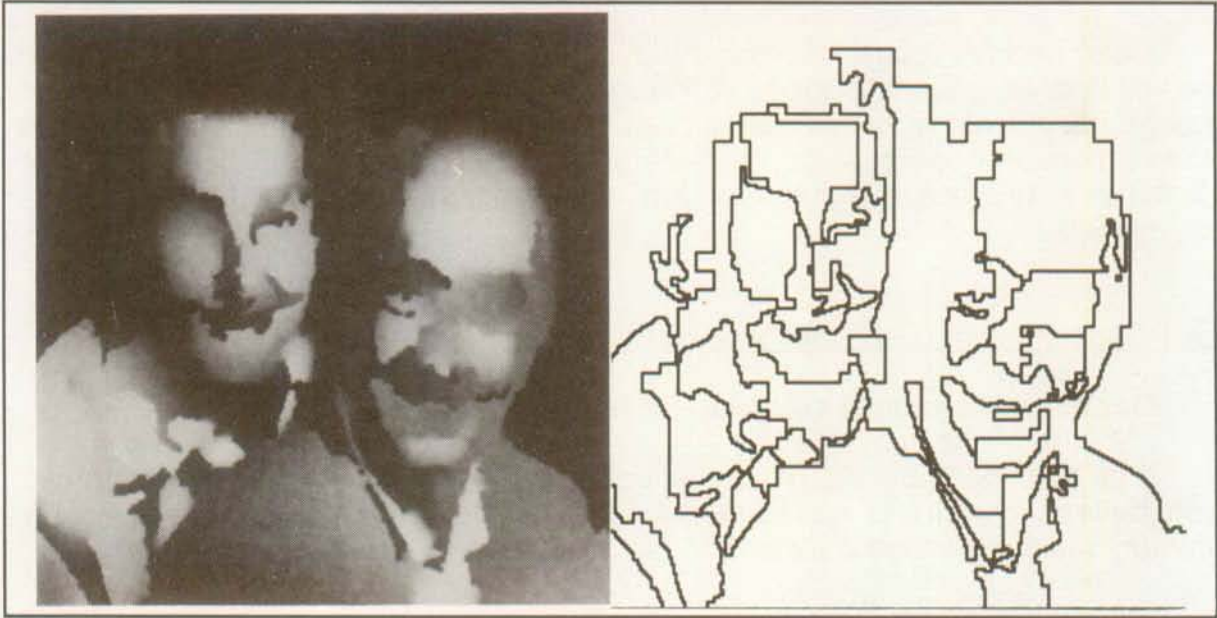
$C_r = 588$ bits (élimination des faux contours dépassant $d=20:n_e=49$); adjonction d'une composante aléatoire.

$C = 11342$ bits

$\rho = 46.23$

Figure 6.45: Image du couple reconstruite après segmentation en 49 régions.

En comparant les résultats de la figure 6.45 et ceux de la figure 6.41, on confirme les remarques précédentes. Contrairement à ce qu'on pourrait penser, la différence de qualité ne s'atténue pas vraiment avec la diminution du nombre de régions bien que les facteurs de compression se rapprochent.



a) Image reconstruite

$\alpha=3$ $N_r=24$
 $M=38$ $L=24$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 6517$
 $\text{Huff}(N_0,N_1,N_2,N_3)=24$

b) Image interfrontière 4-connexe.

$A=499$ $T=5^2$
 $p=4$ $n_c=4606$
 $N_2=23$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4882$ $\text{Huff}_0 = 1600$

$C_c = 6482$ bits

$b_c = 1.407$ bit par point contour.

$C_t = 1342$ bits

$C_r = 420$ bits (élimination des faux contours dépassant $d=20:n_e=35$); adjonction d'une composante aléatoire.

$C = 8244$ bits

$\rho = 63.60$

Figure 6.46: Image du couple reconstruite après segmentation en 24 régions.

Les remarques faites pour la figure 6.45 restent valables pour la figure 6.46.

Il n'est donc pas avantageux de prétraiter l'image originale pour simplifier les images de contrôle utilisées.

6.4 Améliorations :

Cette section aborde le problème de l'amélioration de la méthode en fonction des résultats obtenus à la section 6.2. Le premier paragraphe propose de modifier le degré du polynôme au cours du processus adaptatif de segmentation. Le second montre comment estimer de manière locale à chaque région la composante aléatoire à rajouter aux signaux luminances approximations par des fonctions polynomiales.

6.4.1 Segmentation adaptative avec modification du degré du polynôme:

Des discussions de la section 6.2, il ressort que:

- la segmentation adaptative effectuée avec des polynômes de degré 0 (constantes), garantit la meilleure précision des frontières de régions avec les contours réels de l'image traitée.

- plus on augmente le degré du polynôme, plus on peut suivre des surfaces à variation complexe. On peut dès lors représenter par une seule région des zones homogènes qui devraient être séparées en plusieurs parties si l'on utilise des polynômes de degré inférieur. En arrêtant ainsi le processus de rassemblement à un nombre de régions donné, une segmentation effectuée avec des polynômes de degré élevé garantit de préserver le plus d'objets de la scène originale.

Pour pouvoir conjuguer avantageusement les deux points cités, il faut combiner les degrés des polynômes d'approximation au cours du rassemblement. L'imprécision dans la description des contours réels résulte du problème dû à la singularité de l'approximation au sens des moindres carrés pour des polynômes de degré élevé, problème déjà mentionné au paragraphe 4.5.1.2. Vu les résultats expérimentaux, on peut pallier à cet inconvénient en commençant le processus de segmentation avec constantes. Pour décrire des surfaces à variation complexe avant que l'on ne commence à perdre des détails importants de l'image, il suffit d'augmenter le degré du polynôme à partir du niveau de rassemblement où il n'existe plus de singularité pour décrire le signal luminance de toutes les régions formant la partition de la scène à ce niveau. Le degré du polynôme peut ainsi croître progressivement de 0 à 3.

Pour pouvoir éviter toute singularité, il faut que chaque région de l'image soit formée d'un nombre suffisant de points. Pour cela, il faut éviter de voir apparaître des points isolés au cours du processus de rassemblement. Le critère de dissimilarité entre noeuds du GCR doit donc être EQ (erreur quadratique) ou EQC (erreur quadratique). A partir du moment où l'on atteint des polynômes de degré 3, on peut utiliser pendant quelques étapes de rassemblement le critère d'erreur quadratique courante pondérée (EQCP) pour obtenir des régions de grande taille. On reviendra ensuite au critère de dissimilarité EQ ou EQC.

Pour montrer les performances de la nouvelle méthode de segmentation adaptative avec variation du degré du polynôme, nous avons décidé d'implanter une version légèrement modifiée de celui-ci pour des raisons de simplicité. On procède comme suit:

1.- division de l'image avec des polynômes de degré 0 selon l'algorithme de division décrit au chapitre 3.

2.- rassemblement adaptatif des régions avec des polynômes de degré 0 en utilisant le critère EQC, jusqu'à ce qu'il existe une solution pour faire passer des polynômes de degré 3 par toutes les régions.

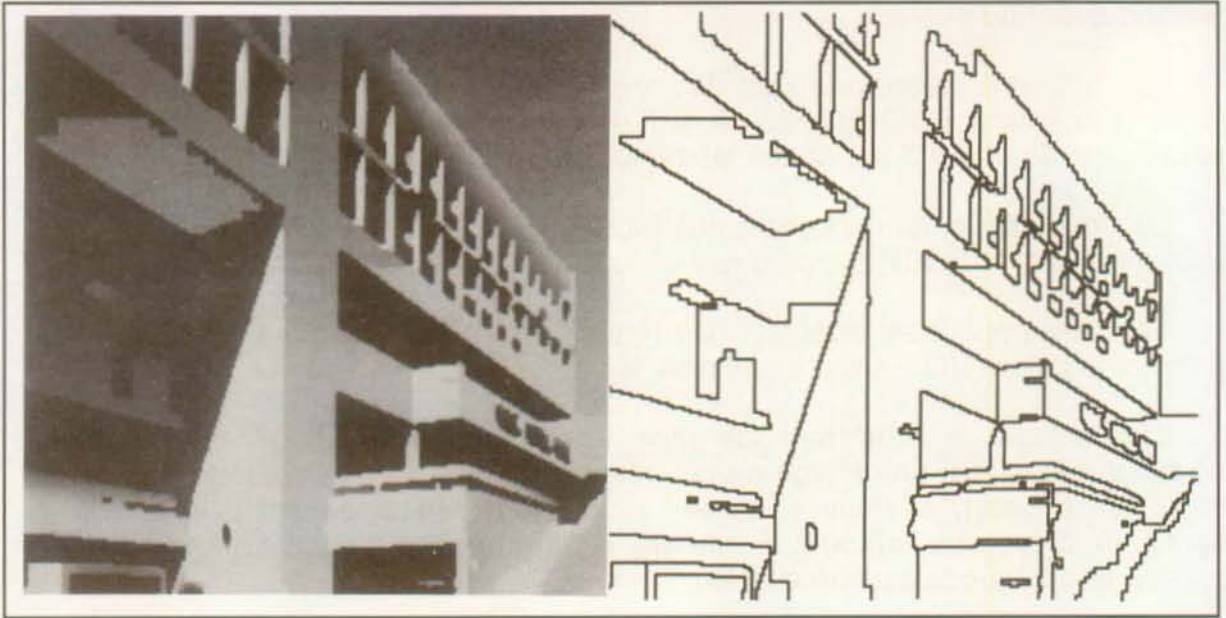
3.- rassemblement adaptatif des régions avec des polynômes de degré 3 en utilisant le critère EQCP pendant quelques étapes de rassemblement.

4.- rassemblement adaptatif des régions avec des polynômes de degré 3 en utilisant le critère EQC jusqu'à atteindre le nombre de régions désiré.

L'avantage de cette méthode apparaît également dans le fait qu'il n'est plus nécessaire d'utiliser une image de contrôle des structures positives et négatives et de devoir séparer les traitements liés aux carrés qui contiennent de ceux qui ne contiennent pas des contours. Seule une image de contrôle des contours doit être utilisée lors du processus de division.

Les figures 6.47 à 6.52 présentent les résultats avec modification du degré du polynôme pour les trois images originales utilisées pour deux valeurs différentes du nombre de régions. On note A_1 le seuil en nombre de régions permettant de passer d'une approximation polynomiale de degré 0 à une approximation polynomiale de degré 3 et A_2 le seuil en nombre de régions pour lequel on passe du critère EQCP (phase 3) au critère EQ (phase 4).

Les figures 6.47 et 6.48 présentent les images reconstruites et les segmentations associées pour l'image du bâtiment formée de 49 et 24 régions respectivement. Les facteurs de compression correspondants sont de 51 à 1 et 68 à 1.



a) Image reconstruite

$N_r=49$ $A_1=99$
 $M=67$ $L=49$
 $N_0=2$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 7620$
 $\text{Huff}(N_0,N_1,N_2,N_3)=58$

b) Image interfrontière 4-connexe.

$A_2=79$ $T=5^2$
 $p=18$ $n_c=5745$
 $N_2=43$ $N_3=3$
 $\text{Huff}(0,2,3,\dots,13) = 5297$ $\text{Huff}_0 = 1900$

$C_c = 7449$ bits

$b_c = 1.297$ bit par point contour.

$C_t = 2804$ bits

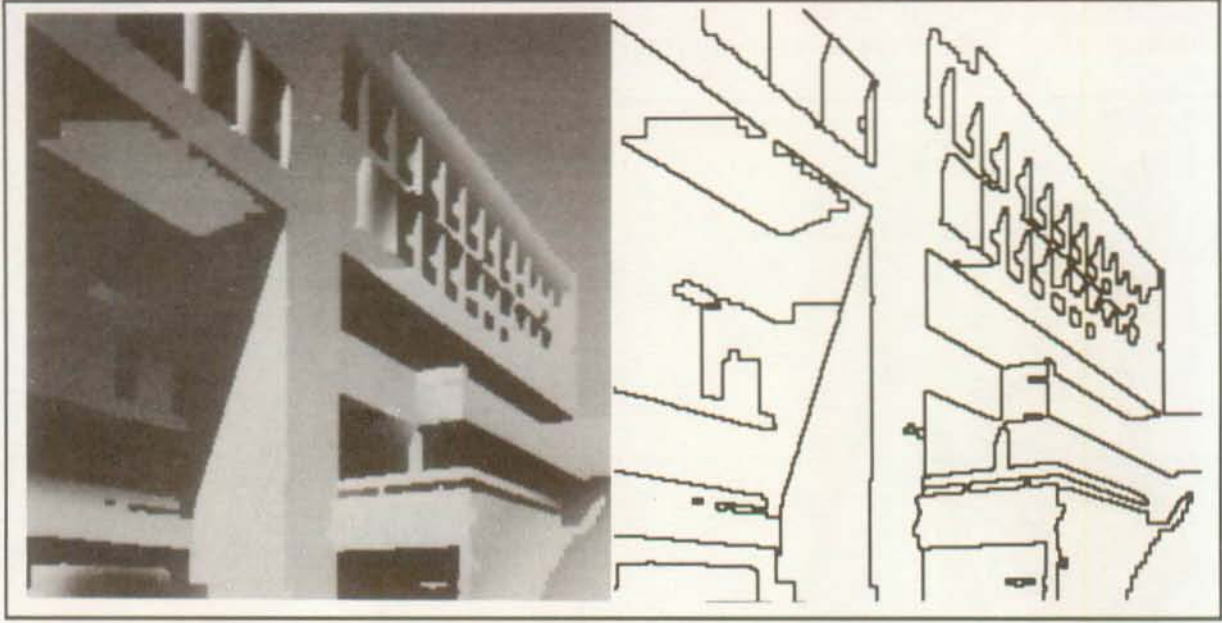
$C_r = 0$ bits (pas d'élimination des faux contours); pas de composante aléatoire.

$C = 10253$ bits

$\rho = 51.13$

Figure 6.47: Image du bâtiment reconstruite après segmentation en 49 régions.

L'image reconstruite de la figure 6.47a est de bien meilleure qualité au niveau de la précision des contours réels que les autres images segmentées en un même nombre de régions. Le défaut apparaissant au sommet de l'arête supérieure droite du bâtiment aurait pu être évité en effectuant un nombre plus faible d'étapes de rassemblement avec le critère EQCP ($A_2 > 79$).



a) Image reconstruite

$N_r=24$ $A_1=99$
 $M=35$ $L=24$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 6422$
 $\text{Huff}(N_0,N_1,N_2,N_3)=45$

b) Image interfrontière 4-connexe.

$A_2=79$ $T=5^2$
 $p=10$ $n_c=4908$
 $N_2=20$ $N_3=3$
 $\text{Huff}(0,2,3,\dots,13) = 4511$ $\text{Huff}_0 = 1627$

$C_c = 6278$ bits

$b_c = 1.279$ bit par point contour.

$C_t = 1428$ bits

$C_r = 0$ bits (pas d'élimination des faux contours); pas de composante aléatoire.

$C = 7706$ bits

$\rho = 68.03$

Figure 6.48: Image du bâtiment reconstruite après segmentation en 24 régions.

La qualité de l'image 6.48a est nettement plus élevée que celle des autres images segmentées en un même nombre de régions. Si l'on compare l'image de la figure 6.35, on peut même observer que la compression est légèrement plus élevée dans ce cas de figure. La phase initiale du processus de segmentation avec des polynômes de degré 0 a permis de garantir la reproduction fidèle des contours réels; l'utilisation de polynômes de degré 3 par la suite évite de devoir associer des zones de l'image ne définissant pas un même espace régulier.

Les figures 6.49 et 6.50 présentent les reconstructions et les segmentations associées pour l'image du caméraman formée de 49 et 24 régions respectivement. Les facteurs de compression correspondants sont de 55 à 1 et de 77 à 1.



a) Image reconstruite

$N_r=49$ $A_1=59$
 $M=69$ $L=47$
 $N_0=2$ $N_1=3$
 $\text{Huff}(1,2,\dots,13) = 6720$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 68$

b) Image interfrontière 4-connexe.

$A_2=49$ $T=5^2$
 $p=12$ $n_c=4390$
 $N_2=42$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 5336$ $\text{Huff}_0 = 1485$

$C_c = 6888$ bits

$b_c = 1.569$ bit par point contour.

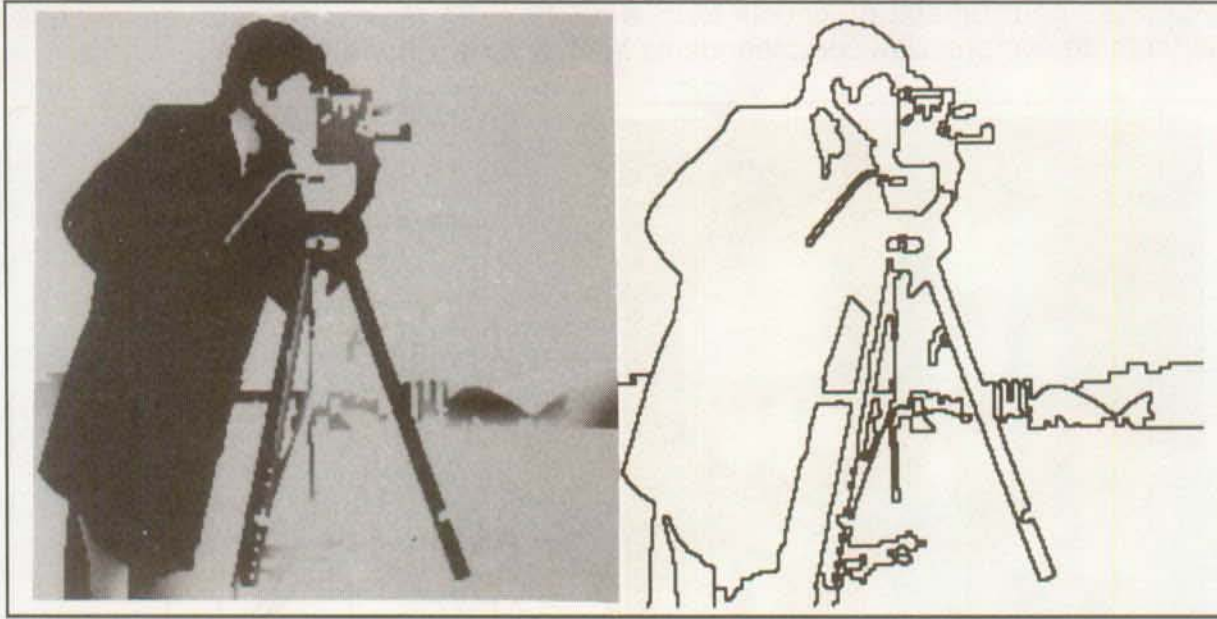
$C_t = 2570$ bits

$C_r = 0$ bits (pas d'élimination des faux contours); pas de composante aléatoire.

$C = 9458$ bits

$\rho = 55.43$

Figure 6.49: Image du caméraman reconstruite après segmentation en 49 régions.



a) Image reconstruite

$N_r=24$ $A_1=59$
 $M=40$ $L=24$
 $N_0=0$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 5342$
 $\text{Huff}(N_0,N_1,N_2,N_3)=24$

b) Image interfrontière 4-connexe.

$A_2=49$ $T=5^2$
 $p=10$ $n_c=4908$
 $N_2=23$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4246$ $\text{Huff}_0 = 1209$

$C_c = 5482$ bits

$b_c = 1.279$ bit par point contour.

$C_t = 1352$ bits

$C_r = 0$ bits (pas d'élimination des faux contours); pas de composante aléatoire.

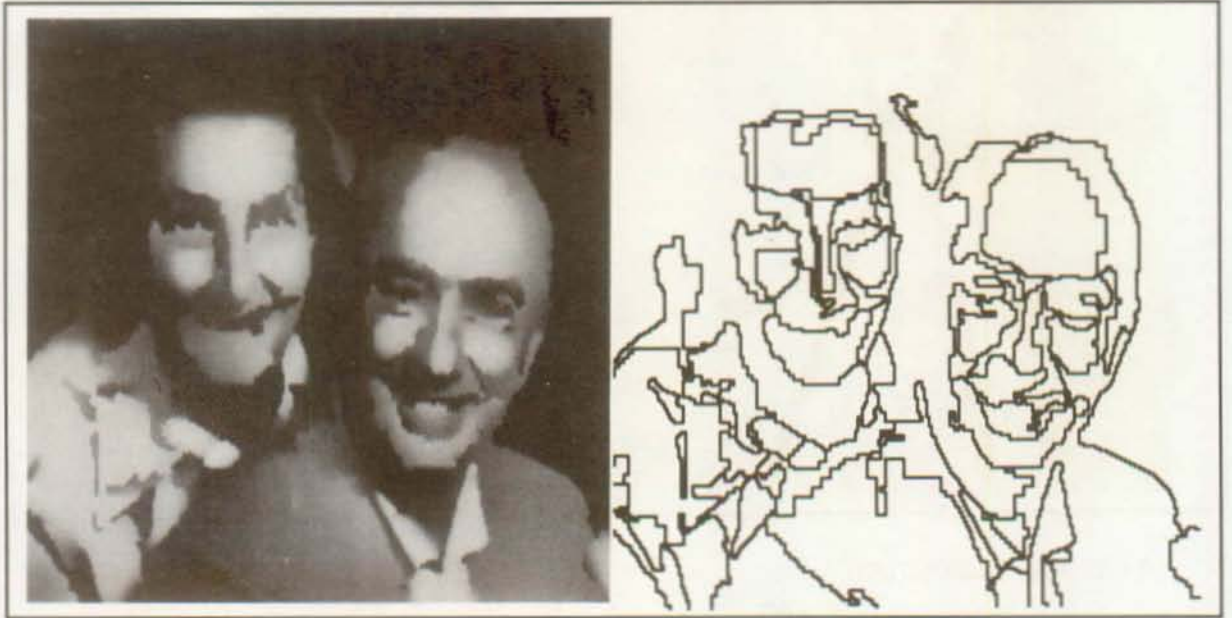
$C = 6834$ bits

$\rho = 76.72$

Figure 6.50: Image du caméraman reconstruite après segmentation en 24 régions.

Les images reconstruites des figures 6.49a et 6.50a sont de qualité comparable à celles des figures 6.9a et 6.10a, qui correspondent aux meilleurs résultats obtenus pour l'image du caméraman à la section 6.2. Les contours réels sont reproduits fidèlement et de nombreux détails sont préservés. On peut remarquer cependant que ces images se distinguent deux à deux par le fait que les segmentations des figures 6.49 et 6.50 présentent une seule grande région correspondant au ciel et au gazon de l'image. Ceci provient de l'utilisation du critère EQCP à un niveau avancé du processus de rassemblement.

Les figures 6.51 et 6.52 présentent les reconstructions et les segmentations associées pour l'image du couple formée de 49 et 24 régions respectivement. Les facteurs de compression correspondants sont de 45 à 1 et de 65 à 1.



a) Image reconstruite

$N_r=49$ $A_1=299$
 $M=67$ $L=49$
 $N_0=1$ $N_1=0$
 $\text{Huff}(1,2,\dots,13) = 8209$
 $\text{Huff}(N_0,N_1,N_2,N_3)=49$

b) Image interfrontière 4-connexe.

$A_2=199$ $T=5^2$
 $p=5$ $n_c=5830$
 $N_2=48$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 6490$ $\text{Huff}_0 = 2082$

$C_c = 8279$ bits

$b_c = 1.420$ bit par point contour.

$C_l = 2763$ bits

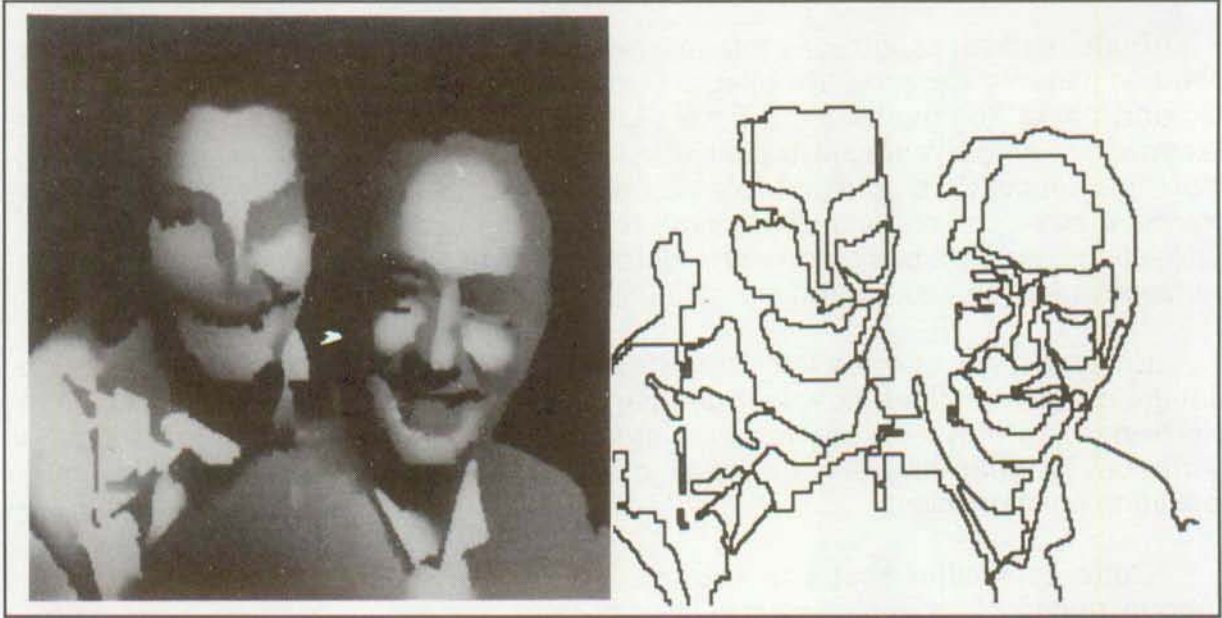
$C_r = 492$ bits (élimination des faux contours; distance $d=20$; $n_e=41$); adjonction d'une composante aléatoire.

$C = 11534$ bits

$\rho = 45.46$

Figure 6.51: Image du couple reconstruite après segmentation en 49 régions.

L'image reconstruite de la figure 6.51a montre une grande précision au niveau des contours réels de la scène ainsi qu'une information détaillée. Si l'on compare cette image à la figure 6.13, il est intéressant de remarquer la différence dans le nombre de régions nécessaires à la reproduction des fronts des personnages.



a) Image reconstruite

$N_f=24$ $A_1=299$
 $M=34$ $L=24$
 $N_0=0$ $N_1=1$
 $\text{Huff}(1,2,\dots,13) = 6278$
 $\text{Huff}(N_0, N_1, N_2, N_3) = 24$

b) Image interfrontière 4-connexe.

$A_2=199$ $T=5^2$
 $p=6$ $n_c=4666$
 $N_2=23$ $N_3=0$
 $\text{Huff}(0,2,3,\dots,13) = 4909$ $\text{Huff}_0 = 1704$

$C_c = 6362$ bits

$C_t = 1322$ bits

$C_r = 324$ bits (élimination des faux contours; $d=20$; $n_e=27$); adjonction d'une composante aléatoire.

$C = 8008$ bits

$\rho = 65.47$

$b_c = 1.279$ bit par point contour.

Figure 6.52: Image du couple reconstruite après segmentation en 24 régions.

Dans le résultat de la figure 6.52, on a perdu l'information de l'oeil gauche de la femme. On peut cependant observer une bonne reproduction des contours réels reconstruits. A la vue de ces résultats et de ceux qui précèdent, on peut en conclure que l'image du couple ne peut être correctement représentée avec un ensemble de 24 régions.

6.4.2 Estimation de la composante aléatoire:

Pour rendre l'aspect des images reconstruites plus naturel, on peut ajouter une composante aléatoire. En effet, si l'on a réduit tant que possible le nombre de régions de la segmentation, on n'a plus que des régions de grande taille où l'approximation polynomiale traduit l'évolution globale du signal de luminance. Il manque l'aspect de granularité locale que peuvent présenter certaines régions. La première idée que l'on peut utiliser pour représenter cette composante aléatoire est l'utilisation d'un bruit blanc gaussien. Si l'on veut limiter la fréquence maximum des variations que l'on veut reproduire, ce bruit peut être filtré passe-bas.

En admettant que les frontières des régions épousent les contours réels de l'image, on peut estimer la valeur des paramètres caractérisant le bruit blanc filtré passe-bas (sa moyenne, sa variance et la fréquence maximale des variations) à partir de la différence entre l'image originale et l'image reconstruite par des fonctions polynomiales.

Cette estimation peut être faite de façon globale sur toute l'image ou pour chaque région prise séparément. On aura dans le second cas une composante aléatoire différente pour chaque région au prix d'une augmentation de l'information à coder.

Les paramètres de bruit peuvent être obtenus à partir des estimateurs de moyenne m^{\wedge} et de variance $(\sigma^2)^{\wedge}$:

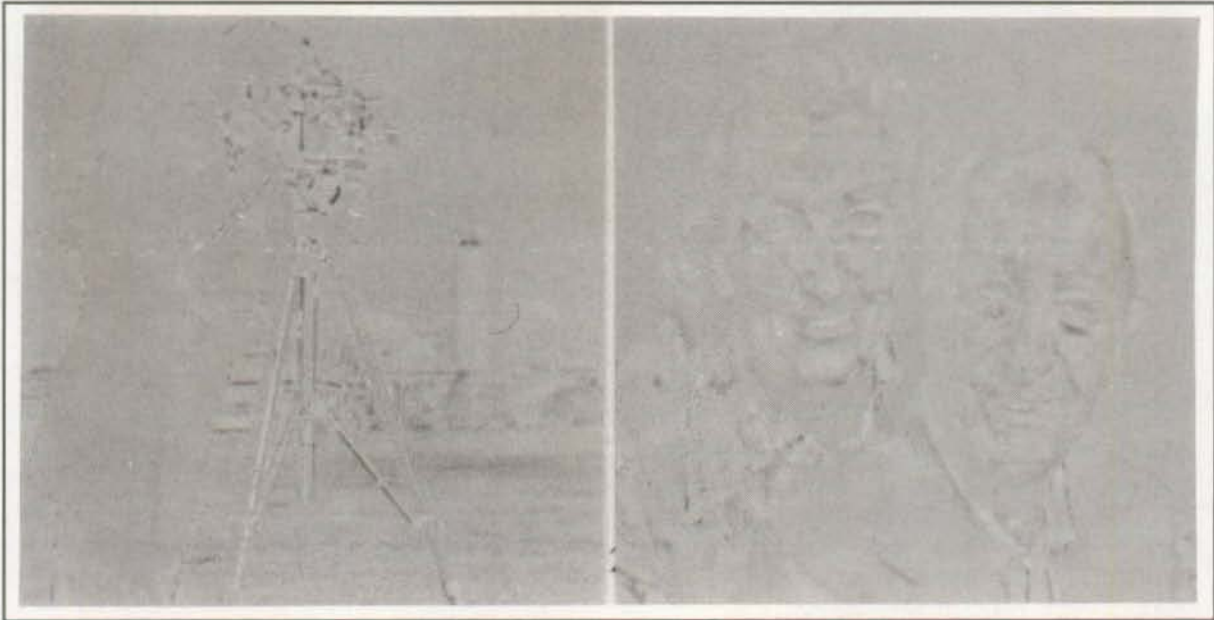
$$m^{\wedge} = (1/N) \sum_{(k,l) \in D} e(k,l) \quad (6.6)$$

$$(\sigma^2)^{\wedge} = (1/(N-1)) \sum_{(k,l) \in D} (e(k,l) - m^{\wedge})^2 \quad (6.7)$$

où D représente le domaine sur lequel on effectue l'estimation, $e(k,l)$ correspond au signal différence au point (k,l) et N est le cardinal du domaine D . D correspond à toute l'image si l'on désire rajouter une composante aléatoire unique pour toute l'image. Il correspond à une région de l'image si la composante est ajoutée localement. En supposant les échantillons du signal différence indépendants, le biais et la variance de ces estimateurs décroît de manière inversement proportionnelle au nombre des échantillons. La variance de l'estimateur de moyenne est une fonction de la variance du signal de différence. La variance de l'estimateur de variance est une fonction de la variance du carré du signal de différence. Comme on peut faire difficilement des hypothèses concernant ce dernier, on ne peut que diminuer la qualité de l'estimation de manière relative. Nous admettrons donc de ne pouvoir obtenir une estimation satisfaisante que lorsque N dépasse 1000. Ainsi, si l'on décide d'estimer les paramètres du bruit localement à chaque région, on ne pourra considérer que les régions contenant plus de 1000 points.

Les valeurs estimées des paramètres de moyenne et d'écart-type du bruit sont majorées par rapport à la réalité par le fait que le signal de différence inclut toutes les parties d'objet qui ont disparu au cours de la segmentation. La figure 6.53

illustre à cet effet le signal de différence correspondant aux images reconstruites du caméraman de la figure 6.49a et du couple de la figure 6.51a.



- a) Différence entre l'original et l'image de la figure 6.49a. b) Différence entre l'original et l'image de la figure 6.51a.

Figure 6.53: Exemples d'images de différence.

Pour effectuer donc une bonne estimation des paramètres, il faut donc que les images reconstruites présentent pratiquement tous les objets de l'image. Sur la base de ces considérations, on a obtenu une valeur légèrement supérieure à 10 pour l'estimation de l'écart-type à partir de la différence des images du bâtiment, du photographe et du couple segmentées en 99 régions avec des polynômes de degré 0. Cette valeur a été fixée à 10 pour tenir compte des erreurs dues à la perte de certains objets de la scène. La fréquence maximale des variations que l'on désire reproduire est difficile à estimer à cause de la nature non parfaitement aléatoire du signal de différence. Avoir des variations de luminance perceptibles au niveau de chaque pixel nous est apparu cependant trop désagréable sur le plan visuel. Aussi avons-nous décidé de moyenniser le bruit blanc gaussien d'écart-type 10 par un filtre moyenneur de taille 3x3. Une taille supérieure (4x4, 5x5...) introduit une composante trop basse-fréquence pour qu'elle puisse être assimilée à un effet de granularité. Le signal de bruit filtré passe-bas est indiqué à la figure 6.54a. Le résultat de son adjonction à l'image du couple de la figure 5.16a est montré à la figure 6.54b.



a) Bruit blanc gaussien ($\mu=0, \sigma=10$) filtré passe-bas par un moyennneur 3x3. b) Image de la figure 5.16a reconstruite après adjonction de la composante aléatoire.

Figure 6.54: Effet de l'adjonction d'une composante aléatoire de reconstruction.

Dans ce qui suit, nous présentons les images reconstruites après adjonction d'une composante aléatoire lorsque les paramètres sont estimés localement à certaines régions de l'image. Ces régions ont été choisies en fonction de leur taille ($N > 1000$) pour avoir une estimation satisfaisante et sur la base de leur importance du point de vue de la sémantique. L'image de la figure 6.55 présente la reconstruction de l'image du couple de la figure 6.51a avec l'adjonction d'une composante aléatoire localement à 6 régions: front de l'homme ($\sigma=15.4$), partie gauche de la veste de la femme ($\sigma=15.3$), arrière-plan de l'image ($\sigma=9.8$), partie droite de la veste de l'homme ($\sigma=15.4$), partie gauche de la veste de l'homme ($\sigma=9.8$), front de la femme ($\sigma=14.7$). Ces valeurs ont été ensuite quantifiées aux niveaux 5 et 10.



a) Composante aléatoire
de bruit blanc gaussien

b) Composante a) filtrée passe-bas
par un moyennneur 3x3.

Figure 6.55: Image du couple de la figure 6.51a reconstruite avec adjonction d'une composante aléatoire locale à 6 régions.

L'image de la figure 6.56 présente la reconstruction de l'image du couple de la figure 6.9a avec l'adjonction d'une composante aléatoire localement à 5 régions: 3 régions correspondant au gazon de l'image ($\sigma=20.8$), manteau du caméraman ($\sigma=19.4$), ciel ($\sigma=12.6$). Les valeurs ont alors été quantifiées aux niveaux 15 et 20. On a préféré l'image de la figure 6.9a à l'image de la figure 6.49a car le ciel et le gazon ne définissent pas une et une seule région.



a) Composante aléatoire
de bruit blanc gaussien

b) Composante a) filtrée passe-bas
par un moyennneur 3x3.

Figure 6.56: Image du caméraman de la figure 6.9a reconstruite avec adjonction d'une composante aléatoire locale à 5 régions.

A notre avis, les images 6.55b et 6.56b légèrement filtrées passe-bas sont de meilleure qualité. Les images des figures 6.55a et 6.56a sont de qualité satisfaisante si on les observe à une certaine distance pour intégrer l'effet de granularité limité au domaine de définition d'un pixel.

7. Conclusion générale

7.1 Principe de la segmentation adaptative :

Les résultats de notre travail ont confirmé le potentiel considérable d'un modèle contour-texture pour les images. La représentation segmentée s'inscrit parfaitement dans le cadre de ce modèle. Une segmentation partage une image en un ensemble de régions dont les frontières correspondent aux contours. Le signal luminance à l'intérieur de chaque région définit sa texture (dans le cadre du modèle utilisé).

Idéalement, une segmentation ne doit présenter que des régions correspondant à des objets réels de la scène. Si l'on veut aboutir à une représentation simplifiée de la scène sans pour autant perdre son contenu sémantique, les propriétés du système visuel humain ont montré qu'il fallait attacher plus d'importance à la représentation des contours. En faisant l'hypothèse que les images naturelles traitées ne sont constituées que par un ensemble de zones homogènes à variation lente, une approximation polynomiale à peu de coefficients est toute adaptée à représenter les surfaces de luminance correspondantes. La méthode de segmentation proposée doit permettre de conjuguer les paramètres de l'approximation polynomiale à la forme des régions. Pour ce faire, l'algorithme de segmentation se décompose en deux étapes adaptatives:

- dans un premier temps, l'image est subdivisée en un ensemble de carrés dont la taille est fixée par une correspondance optimale entre le signal approximé par des polynômes à l'intérieur de chaque carré et l'original. Cette première phase de la segmentation constitue l'opération de division.

- dans un deuxième temps, on construit à partir de l'image dérivée un graphe de contiguïté de régions (GCR) dont les noeuds constituent des régions et les branches lient des régions voisines. Une mesure de dissimilarité est alors associée à chaque branche. Les deux noeuds du GCR liés par la branche présentant le minimum de dissimilarité sont itérativement rassemblés jusqu'à atteindre un nombre de régions donné. Cette deuxième phase de la segmentation constitue l'opération de rassemblement. Pour un GCR initial et une mesure de dissimilarité basée sur les représentations polynomiales de chaque région, on obtient une adaptation optimale des coefficients des polynômes à la forme de chaque région.

7.2 Résultats obtenus dans le contexte du codage d'images :

L'opération de codage consiste à décrire indépendamment chaque région de l'image par sa frontière (information contour) et par l'approximation du signal luminance qui la caractérise (information texture). L'information contour est décrite sans erreur au moyen d'un code de chaîne après avoir légèrement modifié la segmentation de manière à imposer des contraintes à la succession des symboles pouvant exister. On peut ainsi ramener le coût moyen par point contour à 1,5 bit. L'information de texture est caractérisée par des surfaces de reconstruction polynomiales classées par degré de polynôme. On montre comment ramener le coût associé à chaque coefficient de polynômes à 8 bits en transposant leur codage dans le domaine de l'image approximée.

En jouant sur le nombre de régions dont est formée la segmentation on agit directement sur le facteur de compression. La gamme dynamique des compressions va de 25 à 1 à 130 à 1 selon la qualité désirée. Les plus hautes compressions vont de 60 à 1 à 130 à 1 selon le type d'images traitées. En réduisant le nombre de régions, on fait disparaître progressivement des objets ou parties d'objets de l'image segmentées. A partir d'un certain niveau, la distorsion de qualité devient inacceptable ce qui limite les performances en réduction de redondance. Les meilleurs résultats ont été obtenus en faisant croître le degré de polynôme (voir paragraphe 6.4.1) au cours du processus de segmentation et en ajoutant une composante aléatoire aux images reconstruites (voir paragraphe 6.4.2) pour leur donner un aspect naturel.

De tous les résultats présentés, il ressort que l'information contour représente au moins les deux tiers (dans certains cas, les neuf dixièmes) de toute l'information codée. L'augmentation des performances en haute compression ne pourra être obtenue que par une modélisation adéquate de l'information contour, vu qu'on ne peut continuer à réduire le nombre de régions de l'image.

7.3 Problèmes ouverts et améliorations :

Les résultats de la méthode de segmentation adaptative ont été présentés dans le contexte du codage d'images. Les partitions obtenues peuvent par ailleurs servir à interpréter le contenu de la scène. En effet, la segmentation a permis de passer d'une représentation numérique d'une image à une représentation symbolique en termes de régions. Les caractéristiques de ces régions aussi bien en termes de forme, de voisins qu'elles possèdent et du signal luminance qu'elles renferment peuvent être manipulées, comparées... de manière à décrire le contenu de l'image. Il faudra dans un premier temps voir comment il est possible d'élaborer un modèle de connaissances à partir de cette représentation segmentée. Ce qui rend notre technique de segmentation particulièrement attrayante dans ce contexte est le nombre réduit de régions auxquelles on peut ramener la scène de départ.

Le principe de segmentation adaptative peut trouver des applications dans le contexte de la segmentation des textures sans apprentissage. Les textures doivent être prises ici au sens large en tant que zones homogènes de l'image caractérisées par des paramètres statistiques ou déterministes quelconque. Pour appliquer la méthode, il suffit de modifier le modèle d'approximation. Au lieu d'utiliser des fonctions polynomiales, on pourra mesurer différents attributs (moments statistiques, forme d'une règle de placement, forme de primitives).

En ce qui concerne l'amélioration de la méthode de segmentation, on peut essayer d'implanter l'algorithme de la section 6.4.1. sous sa forme la plus générale, c'est-à-dire en caractérisant tout au long du processus de segmentation chaque région par un polynôme de degré 0 à 3 pour autant que celui-ci existe d'une façon univoque (pas de singularité de la matrice S).

En ce qui concerne le codage de l'information contour, il serait intéressant de tolérer des distorsions pour les parties frontières de régions qui ne correspondent pas à des contours réels. Cette méthode ne présente d'intérêt que pour des segmentations formées d'un nombre élevé de régions car ce sont les seules présentant des régions où il peut exister plusieurs frontières correspondant à des

faux contours. On garantit d'augmenter la compression des images reconstruites correspondantes en simplifiant leur description. Le coût nécessaire à marquer les portions de faux contours est faible (environ 500 bits) ainsi qu'on a pu le remarquer lors de l'algorithme de lissage des faux contours (section 5.3).

En ce qui concerne le codage de l'information texture, il faudrait étudier les résultats de la quantification vectorielle en adoptant comme mesure de distance l'erreur quadratique courante au niveau du signal reconstruit. On tient ainsi compte de toutes les valeurs du signal approximé et non seulement des valeurs des coefficients d'approximation.

Enfin, un dernier effort pourra être porté à une meilleure modélisation de la composante aléatoire basée sur les caractéristiques stochastiques de l'image de différence entre l'image approximée par un ensemble de polynômes et l'original. En effet, le modèle de bruit blanc gaussien filtré passe-bas est fort simple.

En conclusion, les problèmes ouverts restent nombreux. Les résultats obtenus montrent l'intérêt de poursuivre la recherche en adoptant un modèle contour-texture pour représenter les images.

Annexe A : Minimisation de la fonction erreur d'approximation :

$$A(\mathbf{e}) = \mathbf{e}^T \mathbf{A} \mathbf{e} \quad (\alpha 1)$$

Chercher le minimum de A revient à chercher le zéro de la dérivée $\partial A / \partial \mathbf{u}$.

$$\begin{aligned} \partial A / \partial \mathbf{u} &= \partial A / \partial \mathbf{e} \cdot \partial \mathbf{e} / \partial \mathbf{u} \\ &= (- \partial A / \partial \mathbf{g}^\wedge) \cdot (- \partial \mathbf{g}^\wedge / \partial \mathbf{u}) \\ &= (\partial ((\mathbf{g} - \mathbf{g}^\wedge)^T \mathbf{A} (\mathbf{g} - \mathbf{g}^\wedge)) / \partial \mathbf{g}^\wedge) \cdot (\partial \mathbf{g}^\wedge / \partial \mathbf{u}) \\ &= (\partial (\mathbf{g}^T \mathbf{A} \mathbf{g} - 2 \mathbf{g}^T \mathbf{A} \mathbf{g}^\wedge + \mathbf{g}^{\wedge T} \mathbf{A} \mathbf{g}^\wedge) / \partial \mathbf{g}^\wedge) \cdot (\partial \mathbf{g}^\wedge / \partial \mathbf{u}) \end{aligned} \quad (\alpha 2)$$

en utilisant le fait que $\mathbf{g}^T \mathbf{A} \mathbf{g}^\wedge = \mathbf{g}^{\wedge T} \mathbf{A} \mathbf{g}$.

Comme $\partial (\mathbf{g}^{\wedge T} \mathbf{A} \mathbf{g}^\wedge) / \partial \mathbf{g}^\wedge = 2 \mathbf{g}^{\wedge T} \mathbf{A}$, (a2) se simplifie

$$\partial A / \partial \mathbf{u} = -2 (\mathbf{g}^T \mathbf{A} - \mathbf{g}^{\wedge T} \mathbf{A}) \cdot (\partial \mathbf{g}^\wedge / \partial \mathbf{u}) \quad (\alpha 3)$$

L'annulation de la dérivée exprimée par (a3) revient à résoudre le système :

$$(\mathbf{g}^{\wedge T} \mathbf{A}) \cdot (\partial \mathbf{g}^\wedge / \partial \mathbf{u}) = (\mathbf{g}^T \mathbf{A}) \cdot (\partial \mathbf{g}^\wedge / \partial \mathbf{u}) \quad (\alpha 4)$$

En prenant la transposée des deux membres, on trouve :

$$(\partial \mathbf{g}^\wedge / \partial \mathbf{u})^T \mathbf{A} \mathbf{g}^\wedge = (\partial \mathbf{g}^\wedge / \partial \mathbf{u})^T \mathbf{A} \mathbf{g} \quad (\alpha 5)$$

ce qui revient finalement à trouver la solution de :

$$(\partial \mathbf{g}^\wedge / \partial \mathbf{u})^T \mathbf{A} \mathbf{e} = 0 \quad (\alpha 6)$$

Annexe B : Modification de l'approximation au sens des moindres carrés par orthogonalisation de l'espace des fonctions de base:

Considérons un ensemble de r fonctions de base Ψ_j . Les r fonctions de base orthogonales Ψ_j' que l'on peut définir sur un domaine D contenant plus de r points s'obtiennent par combinaison linéaire des fonctions Ψ_j :

$$\Psi_j'(x,y) = \sum_{j=1}^r c_{ij} \Psi_j(x,y) \quad (b1)$$

La matrice $r \times r$ correspondant aux éléments c_{ij} est notée \mathbf{C} . La procédure d'orthogonalisation est décrite dans [53]. Si l'espace des fonctions de base Ψ_j' est normé, c'est-à-dire que $\sum_{((k,l) \in D)} \Psi_j'^2(k,l) = 1$, la matrice \mathbf{C} satisfait la relation:

$$\mathbf{C}^T \mathbf{C} = \mathbf{I}_r \quad (b2)$$

où \mathbf{I}_r définit la matrice identité de rang r . En utilisant les notations du chapitre 2, on vérifie aisément que la solution optimale au sens des moindres carrés au moyen de l'ensemble des fonctions Ψ_j' est donnée par:

$$\mathbf{g}'^{\wedge} = \mathbf{Z}'\mathbf{u}' = \mathbf{Z}'(\mathbf{Z}'^T \mathbf{Z}')^{-1} \mathbf{Z}'^T \mathbf{g} \quad (b3)$$

En appliquant la relation (b1) à chaque élément de la matrice \mathbf{Z}' , on a la relation:

$$\mathbf{Z}' = \mathbf{Z} \mathbf{C}^T \quad (b4)$$

Si on remplace \mathbf{Z}' par cette expression dans (b3), on trouve:

$$\begin{aligned} \mathbf{g}'^{\wedge} &= \mathbf{Z} \mathbf{C}^T ((\mathbf{Z} \mathbf{C}^T)^T \mathbf{Z} \mathbf{C}^T)^{-1} (\mathbf{Z} \mathbf{C}^T)^T \mathbf{g} \\ &= \mathbf{Z} \mathbf{C}^T (\mathbf{C} (\mathbf{Z}^T \mathbf{Z}) \mathbf{C}^T)^{-1} \mathbf{C} \mathbf{Z}^T \mathbf{g} = \mathbf{Z} (\mathbf{C}^T (\mathbf{C}^T)^{-1}) (\mathbf{Z}^T \mathbf{Z})^{-1} (\mathbf{C}^{-1} \mathbf{C}) \mathbf{Z}^T \mathbf{g} \\ &= \mathbf{Z} (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{g} = \mathbf{g}^{\wedge} \end{aligned} \quad (b5)$$

où l'on a utilisé successivement le fait que pour deux matrices quelconques $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$ et que pour deux matrices carrées non singulières $(\mathbf{AB})^{-1} = \mathbf{B}^{-1} \mathbf{A}^{-1}$.

La relation (b5) permet de mettre en évidence que l'approximation au sens des moindres carrés reste inchangée après changement de base par transformation linéaire. En particulier, ceci se vérifie pour des ensembles de fonctions qui ont été rendues orthogonales pour le domaine à approximer.

Annexe C: Existence d'une solution polynomiale optimale au sens des moindres carrés sur des domaines carrés de dimension paire:

Considérons un domaine d'approximation rectangulaire contenant respectivement K lignes et L colonnes avec K et L paires. L'existence d'une solution polynomiale au sens des moindres carrés est garantie par la non singularité de la matrice S .

Comme on utilise des fonctions de base exprimées par la relation (2.17), le terme général de cette matrice s'exprime par:

$$\begin{aligned}
 s_{ij} &= \sum_{k=1}^K \sum_{l=1}^L (k^p \cdot l^q) \cdot (k^{p'} \cdot l^{q'}) \\
 &= \sum_{k=1}^K \sum_{l=1}^L (k^{(p+p')} \cdot l^{(q+q')}) = \sum_{k=1}^K k^{(p+p')} \cdot \sum_{l=1}^L l^{(q+q')} \quad (c1)
 \end{aligned}$$

en ayant fait l'hypothèse que $\Psi_j(x,y) = x^p \cdot y^q$ et que $\Psi_j(x,y) = x^{p'} \cdot y^{q'}$.

Du fait de la nature rectangulaire du domaine, chaque terme de la matrice s'exprime donc comme un produit de deux séries d'entiers élevés à une puissance. Si l'on tient compte de l'expression canonique d'un polynôme de degré α exprimée par la relation (2.18), on remarque que les fonctions Ψ_j se succèdent dans un ordre croissant. Il y a successivement 1 terme de degré 0, 2 termes de degré 1, ..., $\alpha+1$ termes de degré α , si bien que la matrice S est de rang $r = (\alpha+1)(\alpha+2)/2$. En effectuant un changement d'origine de telle sorte que la nouvelle origine corresponde au centre de symétrie du domaine rectangulaire, tous les termes de la matrice correspondant à une somme des coordonnées k ou l élevées à une puissance $(p+p'$ ou $q+q')$ impaire sont nuls (en effet, chaque coordonnée positive a son correspondant négatif de même valeur absolue).

En ayant ainsi simplifié considérablement l'expression de la matrice S , il sera montré qu'il existe une solution unique au problème des moindres carrés lorsqu'on utilise des polynômes de degré 0 à 3. Si l'on prend en compte uniquement des carrés, on peut poser $K=L$. La matrice S a alors pour expression dans le cas du degré plus élevé (3):

$$S = \begin{bmatrix}
 a^2 & 0 & 0 & a b & 0 & a b & 0 & 0 & 0 & 0 \\
 0 & a b & 0 & 0 & 0 & 0 & a c & 0 & b^2 & 0 \\
 0 & 0 & a b & 0 & 0 & 0 & 0 & b^2 & 0 & a c \\
 a b & 0 & 0 & a c & 0 & b^2 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & b^2 & 0 & 0 & 0 & 0 & 0 \\
 a b & 0 & 0 & b^2 & 0 & a c & 0 & 0 & 0 & 0 \\
 0 & a c & 0 & 0 & 0 & 0 & a d & 0 & b c & 0 \\
 0 & 0 & b^2 & 0 & 0 & 0 & 0 & b c & 0 & b c \\
 0 & b^2 & 0 & 0 & 0 & 0 & b c & 0 & b c & 0 \\
 0 & 0 & a c & 0 & 0 & 0 & 0 & b c & 0 & a d
 \end{bmatrix} \quad (c2)$$

où $\alpha=K$, $b=\sum_{(i=1,\dots,K)}(i-i_0)^2$, $c=\sum_{(i=1,\dots,K)}(i-i_0)^4$, et $d=\sum_{(i=1,\dots,K)}(i-i_0)^6$. Bien entendu $i_0=(K+1)/2$ correspond à la coordonnée du centre de symétrie du carré.

Dans le cas de polynômes de degré 0, 1 et 2, les matrices **S** correspondantes sont obtenues en prenant les sous-matrices d'ordre 1x1, 3x3 et 6x6 inscrites dans la partie supérieure gauche de la matrice de la formule (c2) respectivement. Pour le cas des polynômes d'ordre 0 et 1, on voit qu'une solution unique existe toujours, les matrices **S** associées étant diagonales. Dans le cas d'un polynôme de degré 2, le déterminant est donné par la relation:

$$|S| = (ab)^4(b^2-ac)^2 \quad (c3)$$

Comme b^2 est toujours plus grand que ac , $|S|$ ne peut jamais s'annuler, ce qui garantit la régularité de la matrice correspondante. De la même manière, dans le cas d'un polynôme de degré 3, le déterminant s'écrit:

$$|S| = (ab)^4(b^2-ac)^4(c^2-bd)^2 \quad (c4)$$

On peut vérifier ici également que c^2 reste toujours plus élevé que bd , ce qui garantit une nouvelle fois la régularité de la matrice **S**. Les expressions des déterminants données par (c3) et (c4) ont été obtenues à l'aide d'un programme de manipulation d'expressions symboliques, en l'occurrence SMP.

Annexe D: Nombre d'opérations nécessaires au calcul des produits matriciels de l'équation (3.12) :

Pour ce faire, nous considérerons le cas d'un carré de dimension txt . Une fois les inverses des matrices de Hankel estimées, la solution optimale implique la multiplication de 5 matrices de taille respectives $r_1 \times r_1$, $r_1 \times t$, txt , $\text{txt} \times r_2$ et $r_2 \times r_2$. Pour simplifier la référence à chacune de ces matrices, nous les noterons \mathbf{M}_1 , \mathbf{M}_2 , ... et \mathbf{M}_5 . De manière à limiter la complexité de calcul, il faut effectuer ces multiplications dans le bon ordre. Pour cela, on utilisera le fait que la multiplication d'une matrice de taille pxq par une matrice de taille qxr requiert pxqxr opérations.

Appelons xy l'indice associé au produit des matrices \mathbf{M}_x et \mathbf{M}_y . On notera par $N(a-b, c-d, e-f, g-h)$ le nombre d'opérations nécessaires aux multiplications matricielles effectuées dans l'ordre indiqué par le quadruplet $(a-b, c-d, e-f, g-h)$, où a, b, c, \dots, h sont des indices.

On a les configurations suivantes:

$$N(1-2, 12-3, 123-4, 1234-5) = r_1^2 t + r_1 t^2 + r_1 r_2 t + r_1 r_2^2 = N_1 \quad (\text{d1})$$

$$N(1-2, 3-4, 34-5, 12-345) = r_1^2 t + r_2 t^2 + r_2^2 t + r_1 r_2 t = N_2 \quad (\text{d2})$$

$$N(1-2, 4-5, 12-3, 123-45) = r_1^2 t + r_2^2 t + r_1 t^2 + r_1 r_2 t = N_3 \quad (\text{d3})$$

$$N(1-2, 4-5, 3-45, 12-345) = r_1^2 t + r_2^2 t + r_2 t^2 + r_1 r_2 t = N_2 \quad (\text{d4})$$

$$N(2-3, 1-23, 123-4, 1234-5) = r_1 t^2 + r_1^2 t + r_1 r_2 t + r_1 r_2^2 = N_1 \quad (\text{d5})$$

$$N(2-3, 1-23, 4-5, 123-45) = r_1 t^2 + r_1^2 t + r_2^2 t + r_1 r_2 t = N_3 \quad (\text{d6})$$

$$N(2-3, 23-4, 1-234, 1234-5) = r_1 t^2 + r_1 r_2 t + r_1^2 r_2 + r_1 r_2^2 = N_4 \quad (\text{d7})$$

$$N(2-3, 23-4, 234-5, 1-2345) = r_1 t^2 + r_1 r_2 t + r_1 r_2^2 + r_1^2 r_2 = N_4 \quad (\text{d8})$$

$$N(2-3, 4-5, 23-45, 1-2345) = r_1 t^2 + r_2^2 t + r_1 r_2 t + r_1^2 r_2 = N_5 \quad (\text{d9})$$

$$N(3-4, 34-5, 2-345, 1-2345) = r_2 t^2 + r_2^2 t + r_1 r_2 t + r_1^2 r_2 = N_6 \quad (\text{d10})$$

$$N(3-4, 2-34, 1-234, 1234-5) = r_2 t^2 + r_1 r_2 t + r_1^2 r_2 + r_1 r_2^2 = N_7 \quad (\text{d11})$$

$$N(3-4, 34-5, 2-345, 1-2345) = r_2 t^2 + r_2^2 t + r_1 r_2 t + r_1^2 r_2 = N_6 \quad (\text{d12})$$

$$N(4-5, 3-45, 2-345, 1-2345) = r_2^2 t + r_2 t^2 + r_1 r_2 t + r_1^2 r_2 = N_6 \quad (\text{d13})$$

De toutes ces configurations ressortent sept nombres d'opérations différents N_1 à N_6 . Le terme $r_1 r_2 t$ est commun à chacun d'eux. Restent 6 autres termes, en l'occurrence $r_1^2 t$, $r_2^2 t$, $r_1 t^2$, $r_2 t^2$, $r_1^2 r_2$ en $r_1 r_2^2$. $N_4 \leq N_1 \leq N_3$ car $t \geq r_2$. De la même façon,

$N_7 \leq N_6 \leq N_2$ car $t \geq r_1$. Entre N_4 et N_7 , tout dépend de la relation entre r_1 et r_2 . Néanmoins, on voit que le nombre minimum d'opérations est obtenu en $N_s = \min(r_1, r_2)xt^2 + r_1xr_2xt + r_1^2r_2 + r_1r_2^2$ opérations. L'ordre dans lequel effectuer les multiplications est donné par le quadruplet associé.

Essayons de montrer maintenant que ce calcul est toujours avantageux par rapport à une formulation non séparable. Dans un tel cas, il faut $N_{ns} = r_1xr_2xt^2 + r_1^2r_2^2$ opérations. Le premier terme correspond au calcul du vecteur h , le second à la multiplication matrice-vecteur $S^{-1}h$. On a cependant négligé le nombre d'opérations relatives à l'inversion de matrice. Ceci prouve les performances de l'approche séparable, car N_{ns} est toujours plus grand que N_s quelle que soit la valeur de t comme il est montré si-dessous. En effet, on peut donc écrire:

$$t^2x\min(r_1, r_2) + r_1xr_2xt + r_1^2r_2 + r_1r_2^2 < r_1xr_2xt^2 + r_1^2r_2^2 \quad (d14)$$

$$t^2(1 - \min(r_1, r_2)/(r_1r_2)) - t + r_1r_2 - r_1 - r_2 > 0 \quad (d15)$$

Le premier terme de cette inégalité est toujours positif, le second toujours négatif et le troisième toujours positif pour $r_1r_2 > 1$. Le membre de gauche de l'inégalité décrit donc une parabole présentant un minimum en $t=1/(2m)$ où m représente le facteur de t^2 dans (d15). La valeur de ce minimum est donnée par:

$$\text{Minimum} = -1/(4m) + r_1r_2 - r_1 - r_2 \quad (d16)$$

On peut montrer qu'il est toujours plus grand que zéro. Posons $\min(r_1, r_2)=r_1$. m vaut dès lors $1-1/r_2$. Etudier le signe du minimum revient à résoudre l'inéquation:

$$4m(r_1r_2 - r_1 - r_2) - 1 > 0$$

$$4r_1r_2 - 4r_1 - 4r_1 + 4r_1/r_2 - 4r_2 + 4 - 1 > 0$$

$$f(r_1, r_2) = 4r_2^2(r_1 - 1) - 4r_1(2r_2 - 1) + 3r_2 > 0 \quad (d17)$$

Or, du fait que $r_1 > 1$ (sinon l'approche séparable ne peut être intéressante), on peut écrire:

$$f(r_1, r_2) < 4r_2^2 - 8r_1r_2 + 4r_1 + 3r_2 \quad (d18)$$

On peut montrer que le terme de droite de cette inéquation présente un minimum absolu en (0.5, 0.875) qui vaut 2.5, ce qui garantit que (d17) reste toujours satisfait.

Annexe E: Expression de l'erreur quadratique lors d'une formulation séparable du problème d'approximation - Application aux polynômes séparables :

Si l'on développe la relation (3.11), l'erreur quadratique est exprimée au moyen de la relation:

$$\begin{aligned}
 EQ &= \text{Trace}((\mathbf{G}^T \mathbf{G} - \mathbf{G}^T \mathbf{G}^\wedge - \mathbf{G}^{\wedge T} \mathbf{G} + \mathbf{G}^{\wedge T} \mathbf{G}^\wedge)) \\
 &= \text{Trace}(\mathbf{G}^T \mathbf{G}) - \text{Trace}(\mathbf{G}^T \mathbf{G}^\wedge) - \text{Trace}(\mathbf{G}^{\wedge T} \mathbf{G}) + \text{Trace}(\mathbf{G}^{\wedge T} \mathbf{G}^\wedge) \\
 &= \text{Trace}(\mathbf{G}^T \mathbf{G}) - 2\text{Trace}(\mathbf{G}^{\wedge T} \mathbf{G}) + \text{Trace}(\mathbf{G}^{\wedge T} \mathbf{G}^\wedge) \quad (e1)
 \end{aligned}$$

en utilisant le fait que $\text{Trace}(\mathbf{A} + \mathbf{B}) = \text{Trace}(\mathbf{A}) + \text{Trace}(\mathbf{B})$, que $\text{Trace}(\mathbf{A}) = \text{Trace}(\mathbf{A}^T)$ et que $(\mathbf{ABC} \dots)^T = (\dots \mathbf{C}^T \mathbf{B}^T \mathbf{A}^T)$. Si l'on explicite le deuxième terme de (e1) à partir de l'équation (3.10), on a:

$$\begin{aligned}
 \text{Trace}(\mathbf{G}^{\wedge T} \mathbf{G}) &= \text{Trace}((\mathbf{V}_k \mathbf{U} \mathbf{V}_l^T)^T \mathbf{G}) = \text{Trace}(\mathbf{V}_l \mathbf{U}^T \mathbf{V}_k^T \mathbf{G}) \\
 &= \text{Trace}(\mathbf{U}^T \mathbf{V}_k^T \mathbf{G}^\wedge \mathbf{V}_l) = \text{Trace}(\mathbf{U}^T \mathbf{H}) \quad (e2)
 \end{aligned}$$

vu que pour toute matrice \mathbf{A} de taille $p \times q$ et \mathbf{B} de taille $q \times p$ $\text{Trace}(\mathbf{AB}) = \text{Trace}(\mathbf{BA})$. Si l'on prend le troisième terme de l'expression (e1), il s'écrit :

$$\begin{aligned}
 \text{Trace}(\mathbf{G}^{\wedge T} \mathbf{G}^\wedge) &= \text{Trace}((\mathbf{V}_k \mathbf{U} \mathbf{V}_l^T)^T (\mathbf{V}_k \mathbf{U} \mathbf{V}_l^T)) \\
 &= \text{Trace}((\mathbf{V}_k \mathbf{U} \mathbf{V}_l^T)^T (\mathbf{V}_k \mathbf{S}_k^{-1} \mathbf{H} \mathbf{S}_l^{-1} \mathbf{V}_l^T)) \\
 &= \text{Trace}(\mathbf{V}_l \mathbf{U}^T (\mathbf{V}_k^T \mathbf{V}_k) \mathbf{S}_k^{-1} \mathbf{H} \mathbf{S}_l^{-1} \mathbf{V}_l^T) \\
 &= \text{Trace}(\mathbf{V}_l \mathbf{U}^T \mathbf{H} \mathbf{S}_l^{-1} \mathbf{V}_l^T) \\
 &= \text{Trace}(\mathbf{U}^T \mathbf{H} \mathbf{S}_l^{-1} (\mathbf{V}_l^T \mathbf{V}_l)) \\
 &= \text{Trace}(\mathbf{U}^T \mathbf{H}) \quad (e3)
 \end{aligned}$$

On remarque donc que dans le cas de l'approximation optimale au sens des moindres carrés, $\text{Trace}[\mathbf{G}^{\wedge T} \mathbf{G}] = \text{Trace}[\mathbf{G}^{\wedge T} \mathbf{G}^\wedge] = \text{Trace}[\mathbf{U}^T \mathbf{H}]$. L'erreur quadratique correspond donc tout simplement à effectuer la différence des énergies des signaux de départ et approximés. Elle s'exprime par:

$$\begin{aligned}
 EQ &= \text{Trace}(\mathbf{G}^T \mathbf{G}) - \text{Trace}(\mathbf{U}^T \mathbf{H}) \\
 &= \left[\sum_{k=1}^K \sum_{l=1}^L g^2(k,l) \right] - \text{Trace}(\mathbf{U}^T \mathbf{H}) \quad (e4)
 \end{aligned}$$

Pour un domaine carré de taille txt , la matrice \mathbf{H} peut être estimée en $N_H = \min(r_1, r_2) \times \text{txt}^2 + r_1 \times r_2 \times \text{txt}$ opérations (voir annexe D). Le calcul de l'énergie des

données approximées s'obtient donc en $N_H + \min(r_1 + r_2^2, r_1^2 + r_2)$ opérations en supposons qu'il faut autant de temps pour effectuer une addition qu'une multiplication. Si $r_1 + r_2^2 > r_1^2 + r_2$, on effectue le calcul de la trace de la matrice $\mathbf{U}^T \mathbf{H}$ sans quoi on calcule la trace de la matrice $\mathbf{U} \mathbf{H}^T$. Le calcul de l'énergie du domaine s'obtient en t^2 opérations. Le calcul de l'erreur quadratique peut se faire de façon optimale en $\min(r_1 + r_2^2, r_1^2 + r_2) + t^2$ opérations si l'on suppose avoir mémorisé lors du calcul des coefficients d'approximation l'expression de la matrice \mathbf{H} .

Pour comparer les performances des méthodes séparables et non séparables, il suffit de comparer le calcul de l'énergie de l'approximation. Dans une formulation non séparable du problème, il faut $r_1 \times r_2$ multiplications et $(r_1 \times r_2) - 1$ additions. Dans le cas séparable, il suffit de $\min(r_1^2, r_2^2)$ multiplications et $\max(r_1, r_2) - 1$ additions. Quelles que soient les valeurs de r_1 et r_2 , la deuxième solution est plus performante.

Annexe F: Modification de la liste des voisins après rassemblement:

Considérons deux noeuds n_i et n_j impliqués dans le rassemblement avec $n_i < n_j$. n_i et n_j possèdent leur ensemble de voisins V_i et V_j , respectivement. Appelons $a_i(a_j)$, le nombre de voisins de $n_i(n_j)$ portant un numéro moins élevé que $n_i(n_j)$. Il existe $b_i = \text{Card}(V_i) - a_i$ voisins de n_i portant un numéro plus élevé que n_i . De façon similaire, il existe $b_j = \text{Card}(V_j) - a_j$ voisins de n_j portant un numéro plus élevé que n_j . Après rassemblement, le nouveau noeud porte le numéro n_i . Les informations relatives au noeud n_j sont éliminées. La liste de voisins de n_i contient tous les voisins de n_j qui n'étaient pas ses propres voisins excepté lui-même, autrement dit:

$$V_i \leftarrow (V_i \cup V_j) \setminus \{n_i, n_j\} \quad (f1)$$

Les éléments de cette liste correspondant aux nouveaux voisins de n_i vont avoir une structure différente selon que le numéro qu'ils portent est compris entre n_i et n_j ou pas. Certains pointeurs de cellule utilisés pour accéder à l'information de voisinage d'un certain noeud vont devenir des cellules et vice-versa. Les cellules ou les pointeurs de cellules doivent être déplacer pour préserver l'ordre des voisins.

On définit les étapes suivantes:

- 1.- balancement de l'arbre après suppression du minimum de dissimilarité et de toutes les mesures de dissimilarité liant n_j à des noeuds appartenant à $V_i \cap V_j$.
- 2.- Insertion ordonnée des voisins v_j de n_j n'appartenant pas à n_i avant rassemblement ($v_j \in (V_i \cup V_j) \setminus ((V_i \cap V_j) \cup \{n_i, n_j\})$); élimination des autres voisins de n_j .
- 3.- remplacement de n_j par n_i dans la liste des voisins des noeuds v_j et remplacement de cellules par des pointeurs de cellule et inversement si nécessaire. Ce processus nécessite un réordonnancement des listes de voisins correspondantes.
- 4.- Calcul des nouvelles mesures de dissimilarité et rebalancement de l'arbre binaire balancé.
- 5.- destruction de toutes les informations restantes relatives au noeud n_j de la liste des noeuds du GCR.

Les étapes 1 et 4 impliquent une simple modification du chaînage des branches de l'arbre pour le rebalancer (voir section 4.3.3). En ce qui concerne les points 2 et 3, la procédure à suivre est un peu plus complexe.

Si n_i et n_j n'ont initialement qu'eux-mêmes pour voisins, aucune modification de la liste des voisins de n_i n'est nécessaire. Il en va de même si n_j n'a pour autre voisin que n_i . Par contre, si l'on a à faire à une situation inverse, on associe la liste des voisins de n_j à n_i en remplaçant systématiquement n_j par n_i . Il est cependant nécessaire de changer la structure de la liste de voisins pour tous les v_j compris

entre n_i et n_j . Pour chaque v_j de la liste des noeuds du GCR, la liste des voisins correspondante est réordonnée en déplaçant la cellule relative à n_j . Le pointeur de cellule provenant de la liste des voisins de n_i indiquera cette cellule.

Si n_i et n_j ont chacun plusieurs voisins, on parcourt les deux listes de voisins en parallèle pour comparer leurs éléments respectifs. L'ordonnement de ses listes selon le numéro de chaque voisin garantit une comparaison rapide. On observe les configurations suivantes:

1) Les a_i premiers voisins de n_j sont comparés aux a_i premiers voisins de n_i pour autant que le numéro des voisins de n_j soit plus petit que n_i . Les comparaisons se font via les pointeurs de cellules. On les effectue jusqu'à dépasser ou atteindre un numéro de voisin de la liste des a_i premiers voisins de n_i . S'il n'est pas atteint, cet élément noté e n'existe pas dans la liste et il y est donc inséré. Selon que e soit supérieur ou pas à n_i on crée une cellule ou un pointeur de cellule. Par la même occasion, la liste des voisins de e est modifiée de manière à faire remonter l'élément $e-n_j$ -md à la place $e-n_i$ -md (dans le cas $e < n_i$). Si e apparaît dans les deux listes, on élimine l'information relative à la branche $e-n_j$ du GCR (cellule dans la liste des voisins de e et pointeur de cellule dans la liste des voisins de n_j).

2) Si le numéro du voisin de n_j devient plus grand que n_i , la comparaison continue de façon similaire mais sur les b_i derniers voisins de n_i . S'il est présent dans les deux listes, on procède comme en 1). S'il n'est pas présent dans la liste, ce qui est vérifié dès que l'on atteint un voisin de n_j portant un numéro supérieur au numéro du voisin de n_i ou lorsqu'on a atteint la fin de la liste des voisins de n_i , on insère cet élément juste avant le voisin portant le numéro supérieur au sien ou après le dernier élément de la liste des voisins de n_i en créant cette fois-ci une cellule vu que ce voisin a un numéro supérieur à n_i . En notant une nouvelle fois par e ce voisin, on crée la cellule n_i - e -md. En se plaçant sur le noeud de valeur e , sa liste de voisins aura la cellule $e-n_j$ -md remplacée par un pointeur de cellule qui indique la cellule n_i - e -md. Ce pointeur est déplacé de manière à respecter l'ordre des voisins.

3) Il s'agit maintenant de traiter le cas des b_j derniers voisins de n_j . Ceux-ci doivent être comparés aux b_i derniers éléments de n_i car le numéro du voisin de n_j est nécessairement plus grand que n_i . On adopte la même stratégie qu'en 2) si ce n'est qu'il n'y a pas inversion de cellules par des pointeurs de cellules.

L'union entre les deux listes de voisins peut-être obtenue ainsi rapidement à cause de l'ordre existant entre chaque cellule ou pointeur de cellule des deux listes.

Annexe G: Expression des paramètres d'approximation par changement d'origine:

Considérons un domaine D comprenant N points. Le signal original sur ce domaine définit un vecteur \mathbf{g} . Son approximation \mathbf{g}^{\wedge} à partir d'un ensemble de r fonctions de base $\Psi_j(x,y)$ ($j=1,\dots,r$) est donnée par:

$$\mathbf{g}^{\wedge} = \mathbf{Z} \mathbf{u} \quad (\text{g1})$$

où \mathbf{u} correspond aux r coefficients d'approximation et la matrice Nxr \mathbf{Z} liée à la géométrie du domaine a pour expression :

$$\mathbf{Z} = \begin{bmatrix} \Psi_1(1) & \Psi_2(1) & \dots & \Psi_r(1) \\ \Psi_1(2) & \Psi_2(2) & \dots & \Psi_r(2) \\ . & . & . & . \\ \Psi_1(N) & \Psi_2(N) & \dots & \Psi_r(N) \end{bmatrix} \quad (\text{g2})$$

$\Psi_j(n)$ définit une notation compacte pour indiquer la valeur de la fonction $\Psi_j(x,y)$ au n^{ième} point du domaine D. La solution optimale au sens des moindres carrés satisfait le système (voir chapitre 2):

$$\mathbf{u} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{g} \quad (\text{g3})$$

La matrice rxr $\mathbf{Z}^T \mathbf{Z}$ est notée \mathbf{S} ; le vecteur $\mathbf{Z}^T \mathbf{g}$ est appelé \mathbf{h} . Le système (g3) se simplifie de la façon suivante :

$$\mathbf{u} = \mathbf{S}^{-1} \mathbf{h} \quad (\text{g4})$$

Si l'on effectue un changement d'origine du système en (x_0, y_0) , tout se passe comme si on calculait l'approximation par rapport à une ensemble de fonctions de base $\Psi'_j(x,y) = \Psi_j(x-x_0, y-y_0)$. On peut souvent passer d'un ensemble de fonctions à l'autre par une simple matrice de changement de base \mathbf{C} de dimension rxr. On écrit alors:

$$\Psi'_i(x,y) = \sum_{j=1}^r c_{ij} \Psi_j(x,y) \quad (\text{g5})$$

Si l'on choisit une approximation polynomiale de degré α , ceci est vérifié. Les coefficients c_{ij} sont alors obtenus grâce au binôme de Newton. D'une façon générale, on peut écrire $\Psi_j(x,y) = x^{p(j)} y^{q(j)}$. On a alors:

$$\Psi'_i(x,y) = (x-x_0)^{p(i)} (y-y_0)^{q(i)}$$

$$\begin{aligned}
& \begin{matrix} p(i) & q(i) \\ = \sum_{k=0} [B(p(i),k) x^k.(-x_0)^{p(i)-k}] \sum_{l=0} [B(q(i),l) y^l.(-y_0)^{q(i)-l}] \\ p(i) & q(i) \\ = \sum_{k=0} \sum_{l=0} [B(p(i),k) B(q(i),l) (-x_0)^{p(i)-k}(-y_0)^{q(i)-l} x^k y^l] \end{matrix} \quad (g6)
\end{aligned}$$

Le terme $x^k y^l$ définit la fonction de base $\Psi_j(x,y)$ pour laquelle $k=p(j)$ et $l=q(j)$, les coefficients c_{ij} s'obtiennent par comparaison entre les relations (g5) et (g6). $B(m,n)$ définit simplement la binômiale de m et n . Elle s'exprime donc par la relation:

$$B(m,n) = m!/(n!(m-n)! \quad (g7)$$

Si l'on indique les paramètres de l'approximation dans la nouvelle base par les mêmes lettres munies d'une apostrophe, on peut écrire:

$$\mathbf{Z}' = \mathbf{Z} \mathbf{C}^T \quad (g8)$$

du fait que le vecteur des fonctions de base Ψ' satisfait la relation $\Psi' = \mathbf{C} \Psi$. La solution optimale au sens des moindres carrés satisfait dans la nouvelle base le système:

$$\begin{aligned}
\mathbf{u}' &= (\mathbf{S}')^{-1} \mathbf{h}' \\
&= (\mathbf{Z}'^T \mathbf{Z}')^{-1} \mathbf{Z}'^T \mathbf{g}
\end{aligned} \quad (g9)$$

En remplaçant \mathbf{Z}' par sa valeur dans (g8), on peut exprimer \mathbf{S}' , \mathbf{h}' et \mathbf{u}' en fonction des anciens paramètres d'approximation. On trouve ainsi:

$$\mathbf{S}' = \mathbf{C} \mathbf{Z}^T \mathbf{Z} \mathbf{C}^T = \mathbf{C} \mathbf{S} \mathbf{C}^T \quad (g10)$$

$$\mathbf{u}' = (\mathbf{C} \mathbf{S} \mathbf{C}^T)^{-1} \mathbf{C} \mathbf{Z}^T \mathbf{g} = (\mathbf{C}^T)^{-1} \mathbf{S}^{-1} \mathbf{C}^{-1} \mathbf{C} \mathbf{h} = (\mathbf{C}^T)^{-1} \mathbf{u} \quad (g11)$$

$$\mathbf{h}' = (\mathbf{Z} \mathbf{C}^T)^T \mathbf{g} = \mathbf{C} (\mathbf{Z}^T \mathbf{g}) = \mathbf{C} \mathbf{h} \quad (g12)$$

Ces relations garantissent d'obtenir l'expression des paramètres d'approximation au sens des moindres carrés après changement d'origine en $O(r^2)$ opérations.

Remarque: Cette propriété peut être généralisée à tout ensemble de fonctions de base $\Psi_j(x,y)$ si après changement d'origine, les nouvelles fonctions de base peuvent être obtenues par combinaison linéaire des fonctions de base de départ. Ceci est équivalent à chercher l'existence d'une matrice de changement de base entre les deux ensembles de fonctions.

Annexe H : Invariance de la distance de Mahalanobis par changement d'origine pour des vecteurs de coefficients polynomiaux:

En reprenant la formulation de l'annexe G, on remarque qu'une solution polynomiale au sens des moindres carrés \mathbf{u}' peut être obtenue après changement d'origine à partir de la matrice de changement de base \mathbf{C} et de la solution optimale \mathbf{u} dans le repère initial. On a ainsi:

$$\mathbf{u}' = (\mathbf{C}^T)^{-1} \mathbf{u} \quad (\text{h1})$$

Si l'on analyse les expressions de la matrice de covariance Γ' et du vecteur moyenne \mathbf{u}'_m à partir d'un ensemble de M vecteurs d'apprentissage \mathbf{u}'_i exprimés dans la nouvelle base, on a :

$$\mathbf{u}'_m = E(\mathbf{u}') = \sum_{i=1}^M \mathbf{u}'_i = \sum_{i=1}^M (\mathbf{C}^T)^{-1} \mathbf{u}_i = (\mathbf{C}^T)^{-1} \sum_{i=1}^M \mathbf{u}_i = (\mathbf{C}^T)^{-1} \mathbf{u}_m \quad (\text{h2})$$

$$\begin{aligned} \Gamma' &= E((\mathbf{u}' - \mathbf{u}'_m)(\mathbf{u}' - \mathbf{u}'_m)^T) = E((\mathbf{C}^T)^{-1}(\mathbf{u} - \mathbf{u}_m)(\mathbf{u} - \mathbf{u}_m)^T \mathbf{C}^{-1}) \\ &= (\mathbf{C}^T)^{-1} E((\mathbf{u} - \mathbf{u}_m)(\mathbf{u} - \mathbf{u}_m)^T) \mathbf{C}^{-1} = (\mathbf{C}^T)^{-1} \Gamma \mathbf{C}^{-1} \end{aligned} \quad (\text{h3})$$

Dans la nouvelle base la distance de Mahalanobis entre deux vecteurs de coefficients \mathbf{u} et \mathbf{v} s'exprime par:

$$\begin{aligned} d'_m(\mathbf{u}', \mathbf{v}') &= (\mathbf{u}' - \mathbf{v}')^T (\Gamma')^{-1} (\mathbf{u}' - \mathbf{v}') = ((\mathbf{u} - \mathbf{v})^T \mathbf{C}^{-1}) (\Gamma')^{-1} ((\mathbf{C}^T)^{-1}(\mathbf{u} - \mathbf{v})) \\ &= ((\mathbf{u} - \mathbf{v})^T \mathbf{C}^{-1}) (\mathbf{C}(\Gamma)^{-1} \mathbf{C}^T) ((\mathbf{C}^T)^{-1}(\mathbf{u} - \mathbf{v})) \\ &= (\mathbf{u} - \mathbf{v})^T (\Gamma)^{-1} (\mathbf{u} - \mathbf{v}) = d_m(\mathbf{u}, \mathbf{v}) \end{aligned} \quad (\text{h4})$$

Ainsi, cette mesure de distance reste invariante par rapport au choix de l'origine pour laquelle on a calculé l'approximation polynomiale au sens des moindres carrés. Cette propriété reste satisfaite pour tout ensemble de fonctions de base pour lequel un changement d'origine correspond à multiplier le vecteur des fonctions de base initial par une matrice de changement de base \mathbf{C} .

La mesure d'erreur quadratique entre vecteurs de coefficients estimés selon le critère des moindres carrés ne présente pas un tel avantage. On trouve dans ce cas:

$$\begin{aligned} d'(\mathbf{u}', \mathbf{v}') &= (\mathbf{u}' - \mathbf{v}')^T (\mathbf{u}' - \mathbf{v}') = (\mathbf{u} - \mathbf{v})^T \mathbf{C}^{-1} (\mathbf{C}^T)^{-1} (\mathbf{u} - \mathbf{v}) \\ &= (\mathbf{u} - \mathbf{v})^T (\mathbf{C}^T \mathbf{C})^{-1} (\mathbf{u} - \mathbf{v}) \end{aligned} \quad (\text{h5})$$

Les deux erreurs quadratiques $d'(\mathbf{u}', \mathbf{v}')$ et $d(\mathbf{u}, \mathbf{v})$ ne sont identiques que si la matrice de changement de base est orthonormée ($\mathbf{C}^T = \mathbf{C}^{-1}$).

Références :

- [1] M. Kunt, A. Ikononopoulos and M. Kocher, *Second-Generation Image-Coding Techniques*, Proceedings of the IEEE, Vol. 73, No. 4 (April 1985).
- [2] M. Kocher and M. Kunt, *A Contour-Texture Approach to Picture Coding*, in Proceedings ICASSP 82, Paris, France, pp.436-440 (May 2-5, 1982).
- [3] A.K. Jain, *Image Data Compression: A Review*, Proceedings of the IEEE, Vol. 69, pp. 349-389 (March 1981).
- [4] H. Welcher, *Texture Analysis - A Survey*, Signal Processing, Vol. 3, pp. 271-282 (July 1980).
- [5] M. Unser, Caractérisation Statistique de Texture: Application à l'Inspection Automatique, Thèse de Doctorat EPFL No. 534, Département d'Electricité, Ecole Polytechnique Fédérale de Lausanne, Suisse (1984).
- [6] D.N. Graham, Image Transmission by Two-Dimensional Contour-Coding, Ph.D. Thesis, MIT, Dept. of Electrical Engineering (1966).
- [7] A. Ikononopoulos and M. Kunt, *High Compression Image Coding Via Directional Filtering*, Signal Processing, Vol. 8, No. 3, pp. 179-203 (May 1985).
- [8] M. Kunt, M. Bénard and R. Leonardi, *Recent Results in High Compression Image Coding*, IEEE Transaction on Circuits and Systems, Special Issue on Image Processing, To be published (Nov. 1987).
- [9] S. Carlsson, Sketch Based Representations in Image Coding, Ph.D. Thesis, Dept. of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden (December 1986).
- [10] P.J. Burt and E.H. Adelson, *The Laplacian Pyramid as a Compact Image Code*, IEEE Transactions on Communications, Vol. COM-31, pp. 532-540 (April 1983).
- [11] D. Wermser, G. Haussmann, and C.E. Liedtke, *Segmentation of Blood Smears by Hierarchical Thresholding*, Computer Graphics and Image Processing, Vol. 25, No. 2, pp. 133-150 (Feb. 1984).
- [12] R.W. Connors, M.H. Trivedi and C.A. Harlow, *Segmentation of a High Resolution Urban Scene Using Texture Operators*, Computer Graphics and Image Processing, Vol. 25, No. 3, pp. 273-310 (March 1984).
- [13] H. Derin and H. Elliott, *Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 1, pp. 39-55 (January 1987).
- [14] L.G. Roberts, *Machine Perception of three-dimensional Objects*, in Optical and Electro-Optical Information Processing, J.T Tippet et al. (Eds.), MIT Press, Massachusetts (1965).

- [15] B.K. Horn, *The Binford-Horn Line Finder*, MIT AI Memo 285, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge (July 1973).
- [16] D.A. Huffman, *Impossible Objects as Nonsense Sentences*, in Machine Intelligence 6, B. Meltzer and D. Michie (Eds.), Elsevier, Amsterdam/New York, pp. 295-323 (1971).
- [17] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York (1973).
- [18] D.L. Waltz, *Understanding Line Drawings of Scenes with Shadows*, in The Psychology of Computer Vision, P.H. Winston (Ed.), McGraw-Hill, New York (1975).
- [19] A.K. Mackworth, *Interpreting Pictures of Polyhedral Scenes*, Proceedings of the 3rd International Conference on Artificial Intelligence, pp. 556-561 (1973).
- [20] K.P. Horn, *Image Intensity Understanding*, MIT AI Memo 335, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge (August 1975).
- [21] S. Shapiro, *Transformations for the Computer Detection of Curves in Noisy Pictures*, Computer Graphics and Image Processing, Vol. 4, pp. 328-338 (1975).
- [22] F. O'Gorman and M.B. Clowes, *Finding Picture Edges Through Collinearity of Feature Points*, Proceedings of the 3rd International Conference on Artificial Intelligence, pp. 543-555 (August 1973).
- [23] S.W. Zucker, R.A. Hummel, and A. Rosenfeld, *Applications of Relaxation Labelling to Line Drawing and Curve Enhancement*, Technical Report 419, Computer Science Center, University of Maryland (1975).
- [24] E.C. Freuder, *Affinity: A Relative Approach to Region Finding*, Computer Graphics and Image Processing, Vol. 5, pp. 254-264 (1976).
- [25] S.L. Horowitz and T. Pavlidis, *Picture Segmentation by a Tree Traversal Algorithm*, Journal of American Computing Machinery, Vol. 23, pp. 368-388 (1976).
- [26] A.M. Nazif and M.D. Levine, *Low Level Image Segmentation : An Expert System*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 5, pp. 555-577 (Sept. 1984).
- [27] M. Kocher, Codage d'Images à Haute Compression Basé sur un Modèle Contour-Texture, Thèse de Doctorat EPFL No. 476, Département d'Electricité, Ecole Polytechnique Fédérale de Lausanne, Suisse (1983).
- [28] R. Olhander, Analysis of Natural Scenes, Ph.D. Thesis, Carnegie-Mellon University (April 1975).
- [29] A.R. Hanson, E.M. Riseman and P. Nagin, *Region Growing in Textured Outdoor Scenes*, Proceedings of the 3rd Milwaukee Symposium on Automated Computation and Control, pp. 407-417 (1975).

- [30] D.H. Hubel and T.N. Wiesel, *Brain Mechanisms of Vision*, Scientific American, Vol. 241, pp. 150-162 (Sept. 1979).
- [31] E.M. Riseman and M.A. Arbib, *SURVEY : Computational Techniques in the Visual Segmentation of Static Scenes*, Computer Graphics and Image Processing, Vol. 6, pp. 221-276 (1977).
- [32] M.L. Baird and M.D. Kelly, *A Paradigm for Semantic Picture Recognition*, Pattern Recognition, Vol. 6, pp. 61-74 (1974).
- [33] J.M. Tenenbaum, M.A. Fischler and H.G. Barrow, *Scene Modeling : A Structural Basis for Image Description*, in Image Modeling, A. Rosenfeld (Ed.), Academic Press, pp.371-389 (1980,1981).
- [34] J.C. Simon, *Errors and Uncertainties in Feature Recognition*, Invited Lecture at the Conference on Mathematics and its Applications in Remote Sensing, Essex, England (May 28-30, 1986).
- [35] M. Kocher and R. Leonardi, *Adaptive Region Growing Technique Using Polynomial Functions for Image Approximation*, Signal Processing, Vol. 11, No. 1, pp. 47-60 (July 1986).
- [36] O.J. Morris, M. de J. Lee and A.G. Constantinides, *Graph Theory for Image Analysis : An Approach Based on the Shortest Spanning Tree*, IEEE Proceedings, Vol. 133, Pt- F, No. 2, pp. 146-152 (April 1986).
- [37] T. Pavlidis, Structural Pattern Recognition, Springer, Berlin (1977).
- [38] V. Torre and T.A. Poggio, *On Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 2, pp. 147-163 (March 1986).
- [39] D.C. Marr and E.C. Hildreth, *Theory of Edge Detection*, in Proc. Roy. Soc. London B, Vol. 207, pp. 187-217 (1980).
- [40] J.F. Canny, Finding Edges and Lines, Massachusetts Institute of Technology, Cambridge, Technical Report 720 (1983).
- [41] M. Hueckel, *An Operator Which Locates Edges in Digital Pictures*, Journal of the Association for Computing Machinery, Vol. 18, No. 1, pp. 113-125 (Jan. 1971).
- [42] R.M. Haralick, *Digital Step Edges from Zero Crossings of Second Directional Derivatives*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 1, pp. 58-68 (Jan. 1984).
- [43] T.S. Huang, G.J. Yang and G.Y. Tang, *A Fast Two-Dimensional Median Filtering Algorithm*, 1978 IEEE Computer Society Conference on Pattern Recognition and Image Processing, pp. 128-130 (1978).
- [44] M. Nagao and T. Matsuyama, *Edge Preserving Smoothing*, Computer Graphics and Image Processing, Vol. 9, pp. 394-407 (1979).
- [45] I. Sobel, Camera Models and Machine Perception, Stanford AI Memo 121, Dept. of Computer Science, Stanford University, Stanford, California (May 1970).

- [46] D. Marr and T. Poggio, A Computational Theory of Human Stereo Vision, in Proc. Roy. Soc. London B, Vol. 204, pp. 301-328 (1979).
- [47] E.W. Cheney, Introduction to Approximation Theory, International Series in Pure and Applied Mathematics, Mac-Graw Hill (1966).
- [48] M.J.D. Powell, Approximation Theory and Methods, Cambridge University Press (1981).
- [49] B.N. Sahney, Polynomial and Spline Approximation - Theory and Application, Proceeding of the NATO advanced Study Institute, D. Reidel Publishing Company (1978).
- [50] P.J. Davis, Interpolation and Approximation, Blaisdell Publishing Company (1963).
- [51] M.G. Cox, Practical Spline Approximation, National Physical Laboratory, Division of Information Technology and Computing, Teddington, Middlesex TW 11 OLW, UK (Feb. 1982).
- [52] T.K. Sarkar, K.R. Siarkiewicz and R.F. Stratton, Survey of Numerical Methods for Solution of Large Systems of Linear Equations for Electromagnetic Field Problems, IEEE Transactions on Antennas and Propagation, Vol. AP-29, No. 6, pp. 847-856 (Nov. 1981)
- [53] F. de Coulon, Théorie et Traitement des Signaux, Traité d'Electricité, Vol. 6, Presses Polytechniques Romandes (1984).
- [54] S. M. Selby, Standard Mathematical Tables, The Chemical Rubber Company, 1967.
- [55] M. Eden, M. Unser, and R. Leonardi, Polynomial Representation of Pictures, Signal Processing, Vol. 10, No. 4, pp. 385-393 (April 1986).
- [56] N. Kalouptsidis, G. Carayannis, and D. Manolakis, Fast Algorithms for Block Toeplitz Matrices with Toeplitz Entries, Signal Processing, Vol. 6, No. 1, pp. 77-81 (January 1984).
- [57] A. Rosenfeld, Quadtree Grammars for Picture Languages, IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-12, pp. 401-405 (1982).
- [58] Y. Cohen, M.S. Landy, M. (Misha) Pavel, Hierarchical Coding of Binary Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No.3, pp. 284-298 (May 1985).
- [59] N. Wirth, Algorithms + Data Structures = Programs, Englewood Cliffs (1979).
- [60] A. Noussan, Segmentation adaptative d'images approchée par différents types de fonctions bidimensionnelles, Projet de diplôme, Laboratoire de Traitement des Signaux, Ecole Polytechnique Fédérale de Lausanne (1985).

- [61] W. Gellert, H. Küstner, M. Hellwich und H. Kästner, Petite Encyclopédie des Mathématiques, Editions K. Pagoulatos (1980).
- [62] M. Eden and M. Kocher, *On the Performance of Contour Coding Algorithm in the Context of Image Coding. Part II: Coding a Contour Graph With No Address Assignment*, Signal Processing (To be published).
- [63] T. Pavlidis, *Survey : A Review of Algorithms for Shape Analysis*, Computer Graphics and Image Processing, Vol. 7, pp. 243-258 (1978).
- [64] M. Eden and M. Kocher, *On the Performance of Contour Coding Algorithm in the Context of Image Coding. Part I: Contour Segment Coding*, Signal Processing, Vol.8, No.4, pp. 381-386 (July 1985).
- [65] S. Carlsson, Sketch Based Representations in Image Coding, Ph.D. thesis, TRITA-TTT 8613, Dept. of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden (Oct. 1986).
- [66] M. Kunt, Comparaison de Techniques d'Encodage pour la Réduction de Redondance d'Images Fac-Simile à Deux Niveaux, Thèse de Doctorat no. 183, Département d'Electricité, Ecole Polytechnique Fédérale de Lausanne, Suisse (1974).
- [67] J. Makhoul, S. Roucos and H. Gish, *Vector Quantization in Speech Coding*, Proceedings of the IEEE, Vol. 73, No. 11, pp. 1551-1588 (Nov. 1985).
- [68] H. Dudley, *Phonetic Pattern Recognition Vocoder for Narrow-Band Speech Recognition*, Journal of Acoustics Society of America, Vol. 30, No. 8, pp. 733-739 (Aug. 1958).
- [69] B. Ramamurthi and A. Gersho, *Classified Vector Quantization of Images*, IEEE Transactions on Communications, Vol. COM-34, No. 11, pp. 1105-1115 (Nov. 1986).
- [70] M. Goldberg, P.R. Boucher and S. Shlien, *Image Compression Using Adaptive Vector Quantization*, IEEE Transactions on Communications, Vol. COM-34, No. 2, pp. 180-187 (Feb. 1986).
- [71] M. Goldberg and H. Sun, *Image Sequence Coding Using Vector Quantization*, IEEE Transactions on Communications, Vol. COM-34, No. 7, pp. 703-710 (Jul. 1986).
- [72] E.W. Forgy, *Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications*, Biometrics, Vol. 21, pp. 768, abstract, (1965).
- [73] M. R. Anderberg, Cluster Analysis for Applications, New York, NY:Academic Press (1973).
- [74] S. Carlsson, *Image Reconstruction from Coded Edge Data*, Proceeding du 1^{er} colloque Image (Traitement, Synthèse, Technologie et Applications), Biarritz, France, pp. 71-77 (Mai 1984).

[75] W.J. Hemmerle, Statistical Computations on a Digital Computer, Blaisdell, New York (1967).

Curriculum Vitae

Riccardo LEONARDI est né à Genève (Suisse) le 26 janvier 1962. Originaire d'Italie, il a effectué sa scolarité à l'Ecole Primaire Française de Genève puis à l'Institut Florimont (Genève). En juillet 1979, l'Académie de Grenoble lui décerne le Baccalauréat scientifique de type C avec la mention Bien.

Il a étudié à l'Ecole Polytechnique Fédérale de Lausanne (EPFL), où il obtint en janvier 1984 son Diplôme d'Ingénieur Electricien EPFL. Il travaille actuellement en tant que collaborateur scientifique d'enseignement et de recherche au Laboratoire de Traitement des Signaux de l'EPFL.

Il a passé une année à l'Université de Carnegie-Mellon (Pittsburgh, USA) dans le cadre d'un programme d'échange durant l'année académique 1981 à 1982.

En 1985, il effectua une étude de faisabilité pour l'Industrie Suisse sur la reconnaissance automatique par voie visuelle des signatures.

Ses intérêts de recherche actuels s'orientent vers le Traitement des Images (Codage d'Images et Analyse de Scène, en particulier), la Reconnaissance des Formes et L'Intelligence Artificielle.

Il est auteur et coauteur d'une dizaine de publications dans diverses revues scientifiques internationales et Proceedings.

